



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Janne Hietala

TIETOKANTAA KÄYTTÄVÄN VERKKOSIVUN LUOMINEN

Liiketalous ja matkailu
2011

TIIVISTELMÄ

Tekijä	Janne Hietala
Opinnäytetyön nimi	Tietokantaa käyttävän verkkosivun luominen
Vuosi	2011
Kieli	suomi
Sivumäärä	38 + 20 liitettä
Ohjaaja	Mika Tamminen

Tämän opinnäytetyön tarkoituksena on antaa kuva siitä, kuinka tietokantaa käyttävä verkkosivu luodaan. Työskentelyn vaiheet käydään läpi yksi kerrallaan, painautuen yksityiskohtaisesti tuloksen kannalta olennaisiin seikkoihin. Tietokannan toteutustyyli esitellään ja koodista nostetaan esille valikoituja osia.

Opinnäytetyön aikana esittelen lyhyesti käyttämäni ohjelmointikielet ja ohjelmistot. Kaikki nämä ovat yksityiskäyttäjien käyttöön tarkoitettuna ilmaisia ja näin ollen jokaisen saatavilla. Tärkeimmässä roolissa ovat tietokannan hallintajärjestelmä MySQL ja ohjelmointikieli PHP, jolla sivun varsinainen ohjelmointi tapahtuu. Tämän lisäksi luodaan katsaus käytettävyyteen eli siihen, miten sivustosta saadaan mahdollisimman helppo ja mukava käyttää. Näitä käytettävyysoppeja hyödynnetään sivuston ulkoasussa.

Työn tuloksena toteutin verkkosivuston, joka tuo sisältönsä tietokannasta. Loin sivustolle myös ylläpitoalueen, jonka kautta tietokannan tietoa voi lisätä, muokata ja poistaa. Sivusto on ulkoisesti hyvin yksinkertainen ja helppo käyttää, mutta se sisältää kuitenkin runsaasti tietoa ja toimivan järjestelmän tämän tiedon hallitsemiseksi. Työn tulosta on mahdollista hyödyntää sellaisenaan, tai siitä voisi muokata vähällä vaivalla toimivan web-sovelluksen johonkin muuhun käyttöön.

Avainsanat	WWW-ohjelmointi, tietokannat, MySQL, PHP
------------	--

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietojenkäsittely

ABSTRACT

Author	Janne Hietala
Title	Creating a Database-Based Web Page
Year	2011
Language	Finnish
Pages	38 + 20 Appendices
Name of Supervisor	Mika Tamminen

The aim of this thesis was to give an understanding of how a database based web page is created. The different stages of working are examined one by one, with more focus on the most essential ones. The execution style of the database is introduced and selected pieces of code are disclosed.

In the thesis there is a short introduction on the programming languages and software I used. All of these are free for private users and therefore available for everyone. Database management system MySQL and programming language PHP have the most important roles on the production and most of the coding is done by PHP. In addition, an overview of usability was done by determining how to do a web page which is as easy and pleasant to use as possible. This knowledge is used on the web site's appearance.

As the result I created a web page which brings its content from a database. I also generated a maintenance area for the web page where the content of the database can be added, altered or deleted. The web page is very simple in its appearance and easy to use but it contains lots of data and a well-working system to manage it. The result product of the thesis work is ready for use as it is or it could be modified to become a web application for some other purpose with only a little effort.

Keywords	WWW Programming, Databases, MySQL, PHP
----------	--

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	5
1.1	Raportin sisältö ja rakenne.....	5
1.2	Kiekko.tk.....	6
2	OHJELMOINTIKIELET JA OHJELMISTOT.....	7
2.1	HTML & XHTML.....	7
2.2	CSS.....	7
2.3	PHP.....	8
2.4	MySQL.....	8
2.5	Apache.....	9
2.6	phpMyAdmin.....	9
2.7	VMware Player.....	10
3	KÄYTETTÄVYYS.....	11
3.1	Viisi keinoa saavuttaa helpposelkoisuus.....	11
3.2	Täydellinen sivusto.....	13
4	SIVUSTON TOTEUTUS.....	15
4.1	Lähtöasetelma.....	16
4.2	Tietokanta.....	16
4.3	Ulkoasu.....	18
4.4	Koodi.....	20
4.4.1	Perusfunktiot.....	20
4.4.2	Liigan pelaajille tarkoitetut sivut.....	22
4.4.3	Ylläpidon kirjautuminen.....	25
4.4.4	Ylläpidon tunnistautuminen.....	28
4.4.5	Ylläpidon lomakkeet.....	29
4.5	Sivujen kuvaukset.....	32
5	LOPPUTULOS JA JOHTOPÄÄTÖKSET.....	35
	LÄHTEET.....	37
	LIITTEET	

1 JOHDANTO

Lähtiessäni tekemään tätä työtä minulla oli kokemusta yksinkertaisten verkkosivujen luonnista ja ylläpidosta. Tietokantojen kanssa olin ollut tekemisissä lähinnä Microsoft Accessilla koulutöiden yhteydessä. Näiden kahden asian yhdistelmää, tietokantaa käyttävää verkkosivua, en ollut koskaan tehnyt ja siihen liittyvä luomisprosessi oli minulle täysi arvoitus. Olin kyllä työskennellyt aikanaan myös tietokantaa käyttävän verkkokaupan kanssa, mutta sen toiminnasta minulla ei varsinaisesti ollut ymmärrystä.

Opinnäytetyöni ideana on siis selvittää, miten tietokantapohjainen verkkosivu luodaan ja dokumentoida tämä luomisprosessi yksityiskohtaisesti. Oletuksena työn dokumentoinnissa on se, että lukijalla on jonkinlaista kokemusta verkkosivujen kanssa toimimisesta. En siis paneudu niinkään verkkosivuston toteutuksen mahdollistaviin perusasioihin, vaan keskityn seikkoihin, jotka tulivat minulle uutena oppina ja jotka mahdollistavat tietokannan käyttämisen verkkosivun kautta.

1.1 Raportin sisältö ja rakenne

Sivuston toteutuksesta kertovassa osassa esitellään suunnittelemani tietokanta. Tämän rooli sivuston toiminnassa on luonnollisesti suuri ja lisähaasteen sen luomiselle asetti tallennettavan tiedon suuri määrä. Lisäksi perustelen ulkoasun ratkaisuja, poimin sivuston PHP-koodista tärkeitä esimerkkejä ja kuvailen toteuttamani sivujen roolit.

Sen lisäksi että sivusto osaa tuoda tietoa tietokannasta esille, loin siihen verkossa toimivan ylläpito-osion. Näiden ylläpitosivujen avulla onnistuu tietojen lisäys, muokkaus ja poistaminen. Koodiesimerkeissä keskitytään etenkin tämän osan toimimiseen.

Empiirisen osan koostuessa sivun tekemisestä käsitellään työn teoreettisessa osassa aiheeseen liittyviä tekniikoita ja ohjelmistoja. Näiden esittelyllä kerrataan kunakin roolia sivuston luonnissa ja luodaan pohjaa empiirisen puolen lähempään tar-

kasteluun. Lisäksi käyn läpi hieman verkkosivujen käytettävyyttä. Tätä oppia hyödynnetään sivuston sisällön esityksessä ja yleisessä ulkoasussa.

1.2 Kiekko.tk

Kiekko.tk on moninpelattava jääkiekkopeli, jota pelataan Java-tuen sisältävällä Internet-selaimella. Tietokoneen ohjaamia maalivahteja lukuunottamatta kaikki pelaajat ovat ihmisten ohjaamia, siten että yksi käyttäjä ohjaa yhtä pelaajaa. Peliä pelaavat kuukausittain kymmenettuhannet eri käyttäjät. (Pynnönen 2011.)

Valitsin sivustoni aiheeksi Kiekko.tk:n pelaajille tarkoitetun liigan. Kiekko.tk ei sisällä varsinaisesti itsessään sarjoja tai turnauksia, vaan näitä on perustettu käyttäjien toimesta. Suurin ja kunnioitetuin on www.kiekkoliiga.net -osoitteesta löytyvä ytimekkäästi nimetty Kiekkoliiga, joka on ollut toiminnassa vuodesta 2004 ja jonka alaisuudessa pelaavat tälläkin hetkellä sadat eri käyttäjät. Liigat toimivat siten, että ottelut pelataan sovittuun tyyliin Kiekko.tk-sivustolla, jonka jälkeen tulokset kirjataan ylös erillisille liigojen tai turnauksien sivuille.

Liigan ylläpitäminen vaatii runsaasti tietokantojen käyttöä ja näin ollen tämä oli sopiva aihe, jolla sain yhdistettyä verkkosivujen tietokantojen käytön opettelun mielenkiintoiseen harrastukseeni. Vaikka aiheena on Kiekko.tk-liiga, niin lopputulos olisi helposti muunnettavissa esimerkiksi oikean jääkiekko- tai jalkapallosarjan käyttöön.

2 OHJELMOINTIKIELET JA OHJELMISTOT

Mitä taitoja vaaditaan tietokantapohjaisen verkkosivun luomisessa? Seuraavassa käyn läpi käyttämiäni ohjelmointikieliä ja ohjelmistoja. Kaikki nämä ovat saatavilla ilmaiseksi Internetistä, ainakin kun kyseessä on yksityiskäyttäjä ja työkalujen avulla saatu aikaansaannos on voittoa tavoittelematon.

2.1 HTML & XHTML

HTML (HyperText Markup Language) on merkintäkieli, jolla kuvaillaan verkkosivujen sisältöä. Kuvailu tapahtuu merkintätagien avulla, yleensä sijoittamalla haluttu teksti aloittavan ja päättävän tagin väliin. Merkintätagista ja tavallisesta tekstistä koostetaan HTML-dokumentti, jonka selain esittää käyttäjälle verkkosivuna. (W3Schools.com 2011.)

XHTML (eXtensible HyperText Markup Language) on XML-kuvauskieleen pohjautuva HTML:n uudempi versio. Kyseessä on käytännössä täysin samanlainen merkintäkieli kuin HTML muutamalla poikkeuksella. Tärkein näistä on se, että XHTML vaatii sivun kehittäjältä tarkempaa koodia. Tämän tarkkuuden hyödyllisyys tulee ilmi siten, että sivusto toimii paremmin eri selaimissa. (Heinisuo, Rauta 2007, 250–251.)

Tämän opinnäytetyön aikana mainitsen jatkossa monesti HTML-kielen, mutta käytännössä HTML:n tilalla voisi olla myös sana XHTML. Käytän siis sivustoni kehittämisessä XHTML-merkintäkieltä edellä mainitusta kehittäjäystävällisyydestä johtuen, vaikka koodi sinällään toimisi yhtä hyvin HTML-pohjaisenakin.

2.2 CSS

CSS (Cascading Style Sheets) tarkoittaa merkintäjärjestelmää, jonka avulla selaimelle voidaan ehdottaa tyyliohjeita HTML-dokumentin esittämiseen liittyen. Pelkällä HTML-kielellä pystyy myös tekemään tekstinmuotoilua, värien vaihtoa ja muita tyyliin liittyviä toimintoja, mutta ne ovat CSS:ää rajoittuneimpia ja ennen

kaikkea hankalampia ylläpitää. Lisäksi uudemmissa HTML-versioissa ulkoasun muokkausmahdollisuuksia on vähennetty, joten suuntaus on ollut kohti CSS:n käyttöä. (Korpela 2003, 2–3.)

CSS-tyyliohjeet voidaan luoda erilliseen tiedostoon ja käyttää niitä sieltä käsin. Tämä helpottaa sivuston ulkoasun luomista ja muokkausta huomattavasti. Tyyliohjeet voivat kattaa kaiken tekstin muotoilusta esimerkiksi sivun eri osien sijaintiin selainnäkyssä. Kaikkea ei tarvitse kuitenkaan aina erikseen määritellä, vaan ainoastaan ne asiat, joiden haluaa poikkeavan normaalista HTML-dokumentin käsittelytavasta.

2.3 PHP

PHP (PHP: Hypertext Preprocessor) on HTML-kieleen liitettävissä oleva ohjelmointikieli, jolla hoidetaan palvelinpuolen ohjelmointia. PHP-koodin erikoispiirre on siis se, että se käsitellään ja muutetaan HTML-kieleksi palvelimella ja lähetetään vasta sitten käyttäjän nähtäville. Näin ollen itse koodi jää salaiseksi ja ulkopuoliset eivät pääse tarkastelemaan sitä. PHP-tiedoston käsittelyä varten palvelimelle tulee olla asennettuna PHP-ohjelmisto. (Kolehmainen 2006, 3.)

PHP-koodia käytettäessä sivu ei ole muodoltaan HTML-tiedosto, vaan PHP-tiedosto. Itse sivun koodi voi kuitenkin olla täysin samanlainen; selain tulkitsee HTML-koodin samaan tyyliin molemmista tiedostomuodoista. PHP-tiedostossa koodin sekaan lisätään PHP-kieltä, joka siis kuitenkin muuntuu palvelimella HTML-kieleksi ennen käyttäjän selaimelle päätymistä. PHP-koodin tunnistaa siitä, että se kirjoitetaan `<?php` ja `?>` -tagien väliin. PHP:n avulla voidaan esimerkiksi luoda evästeitä tai dynaamista sisältöä sivustolle. (Kolehmainen 2006, 4.)

2.4 MySQL

MySQL (My Structured Query Language) on relaatiotietokantojen hallintajärjestelmä. Relaatiotietokanta tarkoittaa tietokantatyyppiä, jossa tieto on tauluissa ja tämän tiedon käsittelyssä voidaan käyttää taulujen välisiä suhteita eli relaatioita. MySQL on tehokas ja monipuolinen ja sitä voidaan hyödyntää useiden ohjelmointikielten, kuten PHP:n, avulla. MySQL:n avulla voidaan tehdä ja ylläpitää monen-

laisia web-palveluita, kuten esimerkiksi blogeja tai verkkokauppoja. (Heinisuo, Rauta 2007, 37–38.)

2.5 Apache

Apache on maailman suosituin palvelinohjelma. Syynä suosioon on sen tehokkuus ja luotettavuus, sekä moneen muuhun suurena etuna oleva ilmaisuus. Sana palvelin tarkoittaa joko varsinaista verkossa olevaa tietokonetta tai Apachen tapaan tietokoneella olevaa ohjelmaa. Apache on siis tietokoneohjelma, joka vastaanottaa ja lähettää tietoja selaimille. Kun käyttäjä kirjoittaa selaimen jonkun tietyn osoitteen, palvelin vastaa siihen ja toimittaa nähtäville osoitteeseen viittaavan tiedoston, eli verkkosivun. Jos kyseessä on PHP-sivu, tällöin palvelin suorittaa ensin PHP-koodin ja toimittaa sen jälkeen HTML:ksi muutetun sivun selaimelle. (Heinisuo, Rauta 2007, 11–12.)

2.6 phpMyAdmin



Kuva 1. Näkymä tietokannasta phpMyAdminissa.

PhpMyAdmin on sovellus, jolla voidaan käyttää vaivattomasti MySQL-tietokantaa. Sovelluksesta löytyy graafisesti esitetyt toiminnot tietokannan ja sen taulujen luomiseen ja ylläpitoon sekä mahdollisuus kirjoittaa SQL-lauseita. PhpMyAdmin onkin erittäin suosittu apuväline, jolla tietokannan parissa työskentely helpottuu huomattavasti. (Heinisuo, Rauta 2007, 98.)

2.7 VMware Player

VMware Player on virtuaalikoneohjelma. Virtuaalikone tarkoittaa nimensä mukaisesti virtuaalista tietokonetta. Kyseessä on siis tietokoneen sisällä oleva toinen tietokone, käytännössä ohjelma, joka toimii kuin oma itsenäinen tietokone. Siihen voidaan asentaa omat käyttöjärjestelmänsä ja ohjelmansa, jotka toimivat kuin normaalistikin.

Virtuaalikoneesta saatava hyöty on siinä, että se on erittäin turvallinen tapa testata verkkoon tulevia sovelluksia. Palvelimen pystyttäminen omalle koneelle voi aiheuttaa tietoturvariskejä, mutta virtuaalikoneelle asennetulle palvelimelle ei päästä muualta käsiksi. (Heinisuo, Rauta 2007, 46.)

3 KÄYTETTÄVYYS

Sivustoa toteuttaessa on syytä panostaa siihen, että tuloksesta tulee mahdollisimman käyttäjäystävällinen. Tavoitteeni tämän opinnäytetyön suunnitteluvaiheessa ei ollut luoda mitenkään erityisen näyttävää sivustoa, vaan panostaa selkeyteen ja käytettävyyteen.

3.1 Viisi keinoa saavuttaa helpposelkoisuus

Steve Krug kertoo kirjassaan *Älä pakota minua ajattelemaan* viisi yksinkertaista menetelmää, joilla sivusto tehdään mahdollisimman helpposelkoiseksi. Helpposelkoisuus on tärkeää, sillä tavallisesti käyttäjä ei tutki sivua kuin hyvin suurpiirteisesti ja napsauttaa ensimmäistä houkuttelevaa vaikuttavaa linkkiä. Jos mitään kiinnostavaa ei heti löydy, käyttäjä voi turhautua ja pahimmassa tapauksessa poistua sivustolta. (Krug 2006, 21)

1) Selkeän visuaalisen hierarkian luominen.

Mitä tärkeämpi elementti, sitä erottuvampi ja keskeisemmällä paikalla sen tulee olla. Yläreunassa oleva teksti osuu silmään aiemmin kuin alareunassa oleva ja isompi teksti kiinnittää huomion paremmin kuin pieni. Selkeä visuaalinen hierarkia toteutuu myös esittämällä toiminnaltaan samankaltaiset asiat samaan tyyliin. Esimerkiksi sivuston eri sivuille vievät navigointilinkit esitetään tietyllä alueella samannäköisinä. Lisäksi sivun sisällön pitää jakautua selkeästi eri osien alle. Tämä tarkoittaa sitä, että käyttäjälle on selvää, mikä kuuluu mihinkin kokonaisuuteen. Alaotsikot löytyvät pääotsikon alta ja tiettyyn aiheeseen viittaavat linkit eivät ole epäloogisessa paikassa. (Krug 2006, 31–33.)

Käyttäjän tulisi siis ensisilmäyksellä havaita, mitkä sivun osat ovat toiminnaltaan samankaltaisia, mitkä sisältyvät toisen osan alle ja mikä on esillä olevien osien tärkeysjärjestys. Selkeän visuaalisen hierarkian ansiosta käyttäjän ei tarvitse pysähtyä miettimään, miten sivu on tarkoitettu käytettäväksi ja mitä löytyy mistäkin. Käyttö sujuu kuin itsestään ja näin ollen aikaa kuluu ainoastaan sisällön tutkimiseen. (Krug 2006, 31–33.)

2) Yleisten käytäntöjen noudattaminen.

Tietyt sivuston sisällön esitystyylit ovat muodostuneet käytännöksi siitä syystä, että ne ovat olleet parhaiten toimivia ratkaisuja. Tästä syystä näissä yleisissä käytännöissä on hyvä pitäytyäkin, ellei sitten satu keksimään jotain todellakin toimivaa uutta lähestymistapaa. Uuden ratkaisun on oltava kuitenkin omaksuttavissa käyttäjän taholta samassa ajassa kuin vanhakin, tai sitten sen on tarjottava sellaista lisäarvoa, että hetkellinen asian pohdiskelu maksaa itsensä takaisin. (Krug 2006, 34–36.)

Esimerkiksi sivuston eri sivujen välillä liikkumisen mahdollistavat navigaatiolinikit olisi syytä olla käytännöksi muodostuneiden tapojen mukaan esillä (Krug 2006, 60.). "Ihmiset eivät käytä Web-sivustoasi, jos he eivät osaa navigoida siinä (Krug 2006, 51.)."

3) Sivujen jakaminen selkeästi eroteltuihin alueisiin.

Käyttäjän pitää ymmärtää hyvin nopeasti sivustolle saavuttuaan, mistä hän saa näkyville mitään tietoa. Päätös sivuston kiinnostavasta annista tehdään hyvin nopeasti ja näin ollen epäselvästi esitetty tai vaikean reitin takana oleva tieto saattaa jäädä kokonaan löytämättä. (Krug 2006, 37.)

4) Linkkien ilmeneminen.

Linkkien tulee erottua muusta tekstistä ja kuvista. Käyttäjää tulee houkutella napsauttamaan linkkejä ja toisaalta tehtävä selväksi, että tavallista tekstiä ei ole tarvetta ryhtyä napsauttelemaan. Näin ollen tavallisen tekstin ja linkkien väritysten on hyvä erottua, tai ainakin niiden muotoilun tulee olla selvästi toisistaan poikkeavaa. Lisäksi alleviivausta on syytä välttää tavallisessa tekstissä, se kun houkuttelee aina testaamaan, josko kyseessä olisi linkki. (Krug 2006, 37.)

5) Häiriötekijöiden minimointi.

Krug kutsuu sivuston ulkoasun häiriötekijöitä kohinaksi, tarkoittaen tällä visuaalista sekavuutta. Jos käyttäjän eteen tulee liian paljon informaatiota ja varsinkin jos se kaikki informaatio ilmoittaa ulkoisella korostuksella olevansa tärkeää, voi

lopputulos olla enemmänkin luoltaantyyntävä kuin houkutteleva. Asiaan auttaa hillittyjen värien käyttö ja rajoitetun informaatiomäärän näyttäminen samalla ruudulla. (Krug 2006, 38.)

Kohinan vähentäminen onnistuu myös minimoimalla sivulla näkyvän tekstin määrän. Esimerkiksi sivustolla olevan lomakkeen täyttöön liittyvät ohjeet ovat yleensä turhia, sillä niitä ei kukaan kuitenkaan jaksa lukea. Pitkän ohjeistuksen sijasta lomake pitäisikin tehdä niin itsestään selväksi, että sen täyttö onnistuu ilman suurempaa pohdiskelua. (Krug 2006, 47.)

6) Huomioitavaa linkeistä.

Krugin viiden neuvon lisäksi poimin esille tärkeän opastuksen linkeistä.

Linkkien määrästä on olemassa paljon erilaisia mielipiteitä, kuten esimerkiksi se, että jokaiselle sivuston eri sivulle pitäisi päästä kolmella napsautuksella. Napsautuksien määrällä ei kuitenkaan sinällään ole väliä, vaan sillä, miten helposti haluttu asia löytyy. Ajatellaan vaikka seuraavaa: Käyttäjä menee sivulle etsimään jotain tiettyä tietoa. Hän napsauttelee neljää häntä houkuttelevaa linkkiä, ennen kuin hän löytää tarvitsemansa. Toisessa tapauksessa käyttäjä menee sivulle, jossa tarvittava tieto olisi vain kahden napsautuksen päässä. Sinne johtavia linkkejä ei kuitenkaan tahdo löytyä, sillä ne eivät tunnu houkuttelevilta haluttuun asiaan liittyen. Mitä luontevammin ja vähemmällä pohdiskelulla asioiden löytäminen siis sujuu, sen parempi. (Krug 2006, 41.)

3.2 Täydellinen sivusto

Kuten todettua, tavoitteenani oli tehdä mahdollisimman selkeä ja helppokäyttöinen sivusto, jossa luotetaan yksinkertaiseen esitystapaan. Tähän tavoitteeseen oli toki helppo päätyä myös siksi, että en koe osaavani tehdä mitenkään huikaisevaa visuaalista jälkeä. Varsinaisena syynä olivat kuitenkin omat kokemukseni Internetin käyttäjänä. Monet sivustot, joita käytän ahkerasti minua kiinnostavan sisällön sa vuoksi, ovat mielestäni turhan sekavan näköisiä. Koen, että tarvitsemaani tietoa pitää kaivaa esille liian vaikeasti. Mielessä on käynyt, että minä osaisin kyllä tehdä tämän paremmin.

Krug huomauttaa kuitenkin asiaan liittyen, että sivustoja suunnittelevilla ja tekemällä on taakkanaan tietty omiin kokemuksiin perustuva ajattelutapa. Verkkosivujen toteutuksessa luotetaan ratkaisuihin, jotka ovat miellyttäneet muilla sivustoilla käyttäjän ominaisuudessa surffaillessa. Vaikka toisin voisi ajatella, tämä ei kuitenkaan tarkoita sitä, että samat asiat miellyttäisivät kaikkia tai olisivat parhaita ratkaisuja. (Krug 2006, 125–126.)

Jokaisella käyttäjällä on oma tapansa toimia verkossa. Tästä johtuen mitään varsinaista keskivertokäyttäjien toimintaan pohjautuvaa mallia siitä, miltä näyttää ja miten toimii optimaalinen sivusto, ei ole olemassa. Sivuston toteutusta miettiessä tuleekin lähteä miettimään asioita tämän nimenomaisen sivuston kannalta. Kuinka asiat on parasta esittää juuri tällä kyseisellä sivulla, jotta sivun kohderyhmä kokee sivun käyttämisen mieluisaksi? (Krug 2006, 128–129.)

4 SIVUSTON TOTEUTUS

Halusin opetella PHP:n ja MySQL:n käyttöä siitä syystä, että näiden asioiden tunteminen on melkein välttämätöntä, jos aikoo työskennellä verkkosivustojen parissa. Yksinkertainen sivusto ei tarvitse tietokantaa pohjaksi, mutta vähänkin monimutkaisempien vaatimusten kanssa eteen tulee nopeasti ongelmia. Tietokannan käyttäminen mahdollistaa tässä luvussa esittelemäni sivuston toteutuksen etenkin seuraavista syistä johtuen:

1) Tietoa voidaan varastoida käytännössä loputtomasti tietokantaan, josta se taas voidaan tuoda vain yhtä sivupohjaa käyttäen esille.

Jos ylläpitää vaikka erittäin yksinkertaista blogia ja tekee sen ilman tietokantaa, tulee merkinnät jakaa jotenkin. Esimerkiksi kuukausien perusteella tehtävä jako tuottaa uuden tiedoston joka kuukausi. Jos sitten myöhemmin haluaa tehdä jotain muutoksia sivuihin, tulee tämä tehdä jokaiseen tiedostoon erikseen.

Jos sivuston toteutukseen ottaa mukaan tietokannan, jonne blogimerkinnät tallennetaan ja PHP:n, jonka avulla tieto siirtyy selaimen ja tietokannan välillä, on tuloksena erittäin paljon käyttäjäystävällisempi versio. Muutoksia tehdessäkin tarvitsee muokata vain yhtä tiedostoa, ja tulos on nähtävillä jokaista merkintää tutkiessa.

2) Tiedon varastoiminen tietokantaan antaa mahdollisuuden suorittaa tällä tiedolla laskutoimituksia ja tuoda sitä valikoiden esille.

Blogi-esimerkkiä ajatellen tietokantaa käytettäessä jokaiselle merkinnälle voisi luoda vaikka kategorian, jota hyödyntämällä saataisiin jaoteltua erityyppiset merkinnät toisistaan ja luotua sitten hakuja, joiden perusteella esille tuotaisiin vain kaikki tiettyyn kategoriaan kuuluvat merkinnät.

3) Sivuston päivitys voidaan hoitaa kirjautumalla selaimen kautta.

Jos luo sivustoa jonkun muun käyttöön, ei voi lähteä siitä oletuksesta, että hän osaa päivittää sivuja FTP-tiedonsiirtomenetelmää hyväksikäyttäen, saati sitten muokata tarvittavaa päivitystä sivun koodin sekaan.

PHP:n ja MySQL:n yhteistoiminnan ansiosta tietokantaa käyttävälle sivustolle voidaan luoda kirjautumista vaativa osio, jonka kautta pääsee tekemään päivityksiä yksinkertaisten lomakkeiden avulla. Pelkästään käyttäjätunnuksen ja salasanan muistamista edellyttävä päivitysjärjestelmä ei vaadi erityistä tietotekniikan ammattitaitoa käyttäjältään.

4.1 Lähtöasetelma

Valitsin työni aiheeksi siis Kiekko.tk-verkkopelin pelaajille suunnatun sivuston, jossa ylläpidetään pelaajien ja näiden joukkueiden liigaa. Sivuston oleelliset osat ovat sarjataulukko ja pistepörssi, joista pelaajat voivat tarkastaa liigan tilanteen. Jokaisen joukkueen ja pelaajan suoritukset tallennetaan ottelukohtaisesti muistiin myöhempää tarkastelua varten.

Sivuston päivityksestä huolehtivat vastuuhenkilöt. Heille myönnetään tunnukset, joilla pääsee kirjautumaan sivustolla olevaan ylläpito-osioon. Kyseisen osion kautta pystyy lisäämään tietokantaan uusia otteluita ja käyttäjän oikeuksista riippuen muokkaamaan myös uutisia sekä pelaajien ja joukkueiden tietoja.

Tämän luvun pohjustuksessa mainitsemani kolme asiaa ovat hyvin kriittisiä liigan pystytyksen kannalta. Tietoa on erittäin runsaasti, sitä esitetään monella sivulla eriytyliin ja sillä tehdään laskutoimituksia. Selainpohjaisen ylläpidon ansiosta tietojen päivittäminen on sujuvaa, helppokäyttöistä ja tehtävissä usean vastuuhenkilön toimesta.

4.2 Tietokanta

Sivuston vaatima tietokanta osoittautui kohtalaisen monimutkaiseksi. *Joukkue*-, *pelaaja*-, *uutiset*- ja *ylläpito*-taulut eivät vaatineet suurta pohdiskelua sisältönsä perustan osalta. Joukkueiden ja pelaajien ottelukohtaiset tilastot vaativat sen sijaan

runsaasti talletettua tietoa moniin eri tauluihin ja näiden taulujen muodostamisen suunnittelu oli haastavaa. Päädyin lopulta ottelutietojen tallennuksessa kolmeen eri tauluun (*ottelu*, *joukkue_ottelu* ja *pelaaja_ottelu*) ja taulut yhdistävään ottelutunnukseen. Lisäksi toteutuksen edetessä huomasin tarpeen taululle nimeltä *vaiheet*, josta lisää seuraavassa, tietokannan taulujen kuvauksessa. Tietokannan rakennetta selkeyttävä ER-kaavio on nähtävissä liitteessä numero 1.

Uutiset

Muista riippumaton taulu, joka sisältää liigan käyttäjille ja seuraajille julkaistavat uutiset. Esimerkkejä normaaleista uutisista ovat kauden aikatauluun, joukkueiden vastuuhenkilöiden toimintaan ja sääntörikkomuksiin liittyvät ilmoitukset.

Yllapito

Muista riippumaton taulu, jonne tallennetaan vastuuhenkilöiden sivustolle kirjautumisen mahdollistavat tunnukset. Ylläpito-osioon kirjautuessaan admin-tyypin käyttäjä voi vaikuttaa ottelutietoihin. Master-tyypin käyttäjä saa puolestaan tämän lisäksi pääsyn uutisten, sekä pelaaja- ja ottelutietojen muokkaamiseen.

Pelaaja

Jokaisen pelaajan perustietojen tallennukseen käytetty taulu. Jos pelaajan tietoja ei löydy tästä taulusta, ei hänelle voi lisätä ottelukohtaisia tilastoja. Tänne määritellään myös se, mihin joukkueeseen pelaaja kuuluu tällä hetkellä.

Joukkue

Joukkueiden perustiedot, kuten peliasujen väriytykset ja joukkueen kuvaus. Mitään pelaajiin liittyvää tietoa ei tähän tauluun tallenneta. Joukkueesta pitää olla olemassa tiedot tässä taulussa, jotta sille voidaan lisätä ottelutietoja.

Ottelu

Yksi *ottelu*-taulun tietue pitää sisällään informaation yhden ottelun ajankohdasta, vaiheesta (onko kyseessä runkosarjaottelu, vai esimerkiksi finaali), peliin osallis-

tuneista joukkueista ja lopputuloksesta. Lisäksi jokaiselle ottelulle luodaan yksilöllinen ottelutunnus. Ottelutunnus liittää pelaajien ja joukkueiden yksilölliset ottelutiedot tässä taulussa oleviin perustietoihin, kuten pelin ajankohtaan.

Joukkue_ottelu

Vaikka *ottelu*-tauluun tallennetaan osallistuneet joukkueet ja pelin lopputulos, tallennetaan samat tiedot tarkemmin eriteltynä tähän tauluun. Molemmat otteluun osallistuneet joukkueet saavat oman tietueensa, johon kirjataan, pelasiko joukkue kotona vai vieraissa, tuliko pelistä voitto, tappio vai tasapeli ja kuinka monta maalia joukkue teki ja päästi. Näiden tietueiden pohjalta muodostetaan liigan sarjataulukko.

Pelaaja_ottelu

Kaikkien otteluun osallistuneiden pelaajien tilastot tallennetaan tähän tauluun omiin tietueisiinsa. Näistä tilastoista näkyy, kuinka monta maalia ja syöttöä pelaaja on missäkin pelissä tehnyt. Tietueiden pohjalta luodaan muun muassa pelaajan kausikohtaiset tilastot ja liigan pistepörssi.

Vaiheet

Muista riippumaton taulu, joka sisältää ainoastaan kauden eri vaiheissa pelattavien otteluiden nimitykset, eli runkosarjaottelu, puolivälieräottelu, välieräottelu, pronssiottelu ja loppuottelu. Tätä taulua käytetään apuna, kun luodaan uutta ottelua sivuston ylläpito-osion kautta. Ottelun luomisen mahdollistavaan lomakkeeseen haetaan taulussa olevat vaiheiden nimitykset, jolloin niitä voidaan hyödyntää alasvetovalikon vaihtoehtoina.

4.3 Ulkoasu

Sivuston ulkoasu on yksinkertainen ja selkeä. Värimaailma rakentuu valkoisen, sinisen ja tekstin perusvärinä olevan mustan varaan. Valkoinen on taustavärinä, kun taas sininen toimii huomion kiinnittävän logon pohjana ja rajaa sivun sisällön helposti erottuvaksi kokonaisuudeksi. Linkkien perusvärit on vaaleansininen.

Kahdesta tutusta ulkomuodon tyylistä, vasempaan reunaan nojaava tai keskitetty, olen valinnut sivustoni käyttöön jälkimmäisen. Sivun rakenne seuraa myös tuttua kolmen osan kaavaa. Ylimpänä on logo, sen alla navigointilinkit ja alimmaisena itse sivun sisältö.

Pääsivuille johdattavat navigointilinkit ovat isot ja erottuvat. Ne ovat sellaisinaan napsautettavia, eikä niissä ole alasvetovalintoja tai muuta erikoista. Sivusto rakentuu näiden kuuden pääsivun varaan ja alasivuille päästään etenemään niiden kautta. Tähän tyyliin eteneminen on pakollista siitä syystä, että monien alasivujen sisältö määrittyy pääsivuilta napsautettavan tiedon mukaan. Ilman tätä tietoa sivu ei tietäisi, mitä sen pitäisi näyttää.

Jokainen sivu alkaa aina otsikolla, jotta käyttäjä tietää, millä sivulla hän on menossa. Pääsivujen otsikot ovat samat kuin navigointipalkin linkkien nimet, paitsi tulokset ja tilastot -linkkien kohdalla. Tulokset-sivulta löytyvät otsikot *sarjataulukko* ja *ottelut*. Tilastot-sivu on puolestaan otsikoitu *pistepörssiksi*. Näin siksi, että linkkien nimet kuvastavat sivujen sisältöä otsikkoja yleisemmällä tasolla. Pelaajat osaavat yhdistää linkkien nimet etsimäänsä tiettyyn asiaan. Otsikot kuvastavat sitten tarkemmin sivulla näkyvää tietoa.

Sivun sisältöön kuuluvat linkit erottuvat alleviivauksellaan ja pääasiallisella vaaleansinisellä värityksellään. Osa linkeistä on myös mustia, jotta käyttäjä ei tuntisi hukkuvasa linkkien tulvaan ja värimaailma tuntuisi selkeämmältä. Näin välttään Steve Krugin mainitsemalta kohinalta. Samalla sivulla olevat mustat ja vaaleansiniset linkit johdattavat kuitenkin eri osiin sivustoa tai tuottavat erilaisia toimintoja, joten värityksien erot ovat myös loogisia.

Ylläpito-osassa sivuille on lisätty oikeaan yläkulmaan *takaisin*-linkit, joilla pääsee palaamaan edelliselle sivulle. Tämä helpottaa käyttäjän toimintaa ja saa ylläpidon tuntumaan yhtenäiseltä kokonaisuudelta, eikä joukolta irrallisia sivuja. Muuten sivuston kaikille eri sivuille pääsee parilla napsautuksella ja selkeän rakenteen ansiosta käyttäjille riittää navigointipalkin pääsivulinkit.

4.4 Koodi

Sivustoni koodin mallina käytin Rami Heinisuon ja Ilkka Raudan kirjoittamaa kirjaa *PHP ja MySQL - Tietokantapohjaiset verkkopalvelut*. Poimin kirjan esimerkeistä olennaisimmat, minun sivustooni sopivat koodinpätkät, joita sitten muokkasin ja täydensin vastaamaan sivustoni tarpeita. Kirja ei tarjonnut niinkään suoraa vastausta siihen, kuinka pääsen tavoitteeseeni, mutta antoi korvaamatonta opastusta, jonka pohjalta sain lähteä soveltamaan oman kokemukseni ja muiden ohjeiden kanssa.

Koodin toiminnan testauksessa käytin teoriaosuudessa esiteltyä VMware Player -virtuaalikonetta, jonka avulla pystyin tutkimaan keskeneräistä sivustoa ilman tietoturvaongelmista murehtimista.

Seuraavaksi käyn läpi sivustoni koodin olennaisimpia osia.

4.4.1 Perusfunktiot

Kaikki sivuston sivut ovat php-tiedostoja ja niiden koodi alkaa aina kutsumalla *funktiot*-sivua. *Funktiot*-sivu sisältää PHP-koodilla kirjoitettuja funktioita, joille on tarvetta monessa eri yhteydessä. Näin ollen näitä samoja funktioita ei tarvitse kirjoittaa eri sivuille erikseen, vaan ne luodaan ainoastaan *funktiot*-sivulle ja tätä sivua kutsuessa ne tulevat käyttöön. Tämä ratkaisu helpottaa sivuston ylläpitoa pitämällä yksittäisten sivujen koodit selkeämpänä ja lyhyempänä, sekä mahdollistamalla funktioiden muutokset vain yhdessä paikassa useiden sijaan.

Listaus 1. Esimerkki sivun koodin rakenteesta.

```
<?php
require("funktiot.php");

html_alku("Tähän saa kirjoittaa <title>-tagien väliin tulevan otsikon");
    echo "<h1>Otsikko</h1>";
    echo "<p>Tekstiä.</p>";
html_loppu();

?>
```

Jokainen sivu käyttää funktioita *html_alku* ja *html_loppu*. Näistä ensimmäinen sisältää sivun perusrakenteen koodin aloituksen ja jälkimmäinen sen lopetuksen. Pelkästään näitä kahta funktiota kutsumalla saadaan siis näkyville sivun rakenne sisältäen yläpalkkina toimivan kuvan, tämän alla olevan navigointipalkin, sekä itse sivun keskustan sisältöä ympäröivät raamit. Esimerkiksi *uutiset*-sivun koodissa keskitytään siis ainoastaan tuomaan uutiset näkyville. Kaikki muu selaimessa näkyvä on peräisin *funtiot*-sivulta *html_alku* ja *html_loppu* -funktioden avulla. Listaus 1 näyttää esimerkkitapauksen sivun rakenteesta yksinkertaisimmillaan.

Sivustoon kuuluu yhteensä 24 php-tiedostoa sekä ulkoasun määrittelyt sisältävä *tyyli.css*-tiedosto. Näistä php-tiedostoista ainoastaan kaksi eivät ole tekemisissä tietokannan kanssa. Loput tuovat tietokannasta tietoa nähtäville tai antavat mahdollisuuden joko lisätä, muuttaa tai poistaa sitä.

Listaus 2. *yhdista_tietokantaan*-funktio *funktiot.php*:ssa.

```
function yhdista_tietokantaan()
{
    $yhteys = mysql_connect("tietokannan_sijainti", "käyttäjätunnus", "salasana");

    if($yhteys == false)
        exit("Tietokantapalvelimeen yhdistäminen epäonnistui.");

    if(!mysql_select_db("tietokannan_nimi", $yhteys))
        exit("Tietokannan valinta epäonnistui.");

    if(!mysql_query("set names utf8", $yhteys))
        exit("Tietokantayhteyden merkistön asettaminen epäonnistui.");

    return $yhteys;
}
```

Yhteys tietokantaan avataan funktiolla *yhdista_tietokantaan* (listaus 2). Tätä funktiota täytyykin siis kutsua aina ennen kuin halutaan suorittaa SQL-kyselyitä. Kutsuminen tapahtuu yksinkertaisella lauseella "*\$yhteys = yhdista_tietokantaan();*". Tässä lauseessa kutsutaan *yhdista_tietokantaan*-nimistä funktiota ja sijoitetaan funktion antama tulos muuttujaan nimeltä *\$yhteys*. Kun sivulla halutaan jatkossa käyttää tietokantayhteyttä, tarvitsee vain viitata kyseiseen muuttujaan.

4.4.2 Liigan pelaajille tarkoitetut sivut

SQL-lauseiden teko on tuttua kaikille vähänkin tietokantojen kanssa työskennelleille. Vaikka tietojen tuominen esimerkiksi sivustoni sarjataulukkoon tai piste-pörssiin vaativat pitkiä ja monimutkaisia SQL-lauseita, oli niiden tekeminen minulle kuitenkin suhteellisen helppoa jo pelkän aiemman Microsoft Access -kokemukseni perusteella. Kysymys kuuluikin, kuinka SQL-lauseen saa tuomaan tietoa verkkosivulle ja kuinka tätä tietoa voi käyttää.

SQL-kysely luodaan PHP-koodiin *mysql_query()*-funktion avulla. Sulkeiden sisään lisätään ensin lainausmerkein varustettu SQL-lause ja sitten pilkulla eroteltuna aiemmin mainittu *\$yhteys*-muuttuja. *mysql_query*-funktiolle siis kerrotaan, mikä haku suoritetaan ja minne. Funktion antama tulos sijoitetaan muuttujan sisälle. Kun tämän muuttujan käsittelee *mysql_fetch_assoc()*-funktiolla, sijoittamalla sen sulkeiden sisään, saadaan tulokseksi taulukkomuotoinen rivi, jossa on niin monta arvoa kuin SQL-lausekkeessa oli haettuja sarakkeita. Jälleen tulos sijoitetaan muuttujaan, joka voisi olla havainnollistavasti nimeltään vaikka *\$rivi*.

Näin SQL-lauseella haetut sarakkeiden arvot ovat käytettävissä helposti *\$rivi*-muuttujan avulla. Jos esimerkiksi SQL-lause haki *pelaaja*-taulusta kaikki tietyn nimisen pelaajan tiedot, niin tämän pelaajan joukkueen nimi löytyisi *\$rivi*-muuttujasta komennolla *\$rivi["joukkue"]*. Joukkueen nimen tulostus tekstimuotoon onnistuisi siis komennolla *"echo \$rivi["joukkue"];"*, jossa joukkue-sana viittaa *pelaaja*-taulun sarakkeeseen nimeltä *joukkue*.

SQL-lause voi antaa tulokseksi myös monta riviä tietoa. Tällainen haku voisi olla esimerkiksi kaikkien samassa joukkueessa pelaavien pelaajien nimien haku *pelaaja*-taulusta. Tällaisessa tapauksessa *mysql_query*:n antama tulos voidaan käsitellä *mysql_fetch_assoc*:illa while-silmukan sisällä.

Listaus 3. Tietokantahaun tulostus while-silmukan avulla.

```

while($rivi = mysql_fetch_assoc($tulos))
{
    echo $rivi["nimi"];
    echo "<br />";
}

```

Listaus 3:n koodi tuottaisi siis listan joukkueen pelaajien nimistä. Nimien määrä olisi täysin sidoksissa taulun sisältöön. PHP-koodissa jatkettaisiin eteenpäin vasta sitten, kun while-silmukka olisi tulostanut kaikki löytyneet nimet.

Näillä keinoilla onnistuu tiedon hakeminen tietokannasta ja sen käyttäminen sivustolla. Mutta entäs sivujen dynaamisuus? Kun esimerkiksi kaikkien pelaajien yksittäiset tiedot luetaan *pelaaja.php*-sivulta, niin miten saadaan määriteltä se, kenen tiedot näytetään?

Tässä vaiheessa otetaan käyttöön *isset*-funktio. Tämä funktio pystyy poimimaan arvoja sivun latautuessa. Yksi tapa on arvojen hankkiminen *\$_GET*-taulukolla, joka tarkoittaa niiden etsimistä osoiteriviltä. Funktio muodossa "*isset(\$_GET["nimi"])*" tarkoittaisi siis sitä, että osoiteriviltä etsitään arvoa nimeltä *nimi*. Jos arvo löytyy, niin se voidaan muuttaa muuttujaksi. Tämä tehdään lauseella "*\$muuttuja = mysql_real_escape_string(\$_GET["nimi"], \$yhteys);*". *mysql_real_escape_string* käsittelee saadun arvon sellaiseksi, että sitä voi käyttää tietokantahaussa. Kun arvo otetaan osoiteriviltä, niin se tulee käsitellä tällä tavoin. Tämä siksi, että käyttäjähän voi kirjoittaa osoiteriville ihan mitä vain, joten täten varmistetaan, että arvo on käyttökelpoinen.

Kun osoiteriviltä on poimittu arvo muuttujaan *\$muuttuja*, niin sitten sitä voidaan käyttää osana tietokantahakua (listaus 4).

Listaus 4. Muuttujan käyttö osana tietokantahakua.

```

SELECT nimi, joukkue, ppaikka
FROM pelaaja
WHERE nimi = '$muuttuja'

```

Näin haun tulos perustuu osoiterivillä annettuun arvoon. Sivun koodi siis pysyy aina samana, mutta sen näyttämät tiedot muuttuvat osoiterivin mukaan. Miten tuo osoiterivillä oleva arvo sitten määritellään? Jatketaan aiempaa while-silmukka esimerkkiä.

Listaus 5. Pelaajakohtaisten linkkien tulostus while-silmukan avulla.

```

while($rivi = mysql_fetch_assoc($tulokset))
{
    echo " <a href=\"pelaaja.php?nimi=", $rivi["nimi"], "\">";
    echo $rivi["nimi"];
    echo "</a>";
    echo "<br />";
}

```

Listaus 5:n while-silmukka tulostaa listan jokaisen pelaajan nimestä siten, että jokainen nimi on samalla myös linkki. Tämän linkin osoite muodostuu *\$rivi*-muuttujasta otetun arvon mukaan. Jos haun perusteella olisi siis löytynyt esimerkiksi pelaaja nimeltä Hietala, niin listassa tämä nimi näkyisi linkkinä, jonka osoite olisi "*pelaaja.php?nimi=Hietala*". Kun *pelaaja*-sivulla on *isset*-funktio, joka etsii osoiteriviltä *nimi*-arvoa ja vie sen SQL-lauseensa hakuehdoksi, niin tuloksena on Hietala-nimisen pelaajan tietojen näyttö (listaus 6).

Listaus 6. Sivu tulostaa tietokannasta saadut tiedot pelaajasta, jonka nimi on määriteltä osoiterivillä. Jos osoiterivillä olevaa arvoa ei löydy, tulostetaan virheilmoitus.

```
<?php
require("funktiot.php");
$yhteys = yhdistä_tietokantaan();

if(isset($_GET["nimi"]))
{
    $muuttuja = mysql_real_escape_string($_GET["nimi"], $yhteys);
    $tuloks = mysql_query("SELECT nimi, joukkue, paikka
FROM pelaaja WHERE nimi = '$muuttuja'", $yhteys);
    if(!$tuloks)
        exit("Tietokantahaku ei onnistunut!");

    $rivi = mysql_fetch_assoc($tuloks);
    if($rivi == false)
    {
        html_alku("Pelaajasivu");
        echo "<p>Pelaajaa ei löytynyt!</p>";
        html_loppu();
    }
    else
    {
        html_alku("Pelaajasivu");
        echo "<h1>Pelaaja nimeltä", $rivi["nimi"], "</h1>";
        echo "<p>Joukkue: ", $rivi["joukkue"], "</p>";
        echo "<p>Pelipaikka: ", $rivi["paikka"], "</p>";
        html_loppu();
    }
}
else
{
    html_alku("Pelaajasivu");
    echo "<p>Pelaajaa ei ole määritetty!</p>";
    html_loppu();
}
?>
```

Näitä menetelmiä soveltamalla saadaan aikaiseksi kaikki sivuston normaalikäyttäjille näkyvät sivut. Liigan pelaajat haluavat tutkia sivustolta erinäisiä tilastoja ja sivun kehittäjän työ on vain tuoda nämä tiedot heidän nähtävilleen. Kaikkien tietojen tuominen onnistuu valitsemalla *isset*-funktiolla käyttäjän haluama kohde, viemällä tämä kohde muuttujana SQL-lauseeseen ehdoksi ja tuomalla sitten SQL-lauseen tulos näkyville.

4.4.3 Ylläpidon kirjautuminen

Sivuston ylläpito-osiossa tuli olla mahdollisuus sisäänkirjautumiselle, jonka jälkeen avautuu pääsy muokkaamaan erinäisiä tietokannan tietoja.

Sisäänkirjautuminen toteutetaan luomalla täytettävät käyttäjätunnus ja salasana - kentät HTML:n `<form>` ja `<input>` -tagien avulla. Tämä lomakkeen luominen on aikalailla HTML:n peruskäyttöä, joten en paneudu siihen, vaan lomakkeen lähettämisen tiedon käsittelyyn.

Kun kentät täytetään ja napsautetaan kirjaudu-painiketta, niin selain ryhtyy käymään uudestaan sivun koodia läpi. Tällä kertaa sivulla käytetään *isset*-funktion `$_POST`-muotoa, joka poimii lomakkeella lähetettyjä tietoja. Mikäli funktio *isset(\$_POST["kirjaudu"])* löytää kirjaudu-napin painamisella lähetetyn *kirjaudu*-arvon, niin sivu ryhtyy tarkistamaan käyttäjätunnusta ja salasanaa.

Salasanaa ei voi tallentaa tietokantaan sellaisenaan, vaan se pitää muuttaa tunnistamattomaan muotoon. Tämä siksi, että vaikka joku ulkopuolinen pääsisi tunkeutumaan tietokantaan, ei hän saisi käyttäjien salasanoja selville. Salasanan muutos tehdään SHA-1-algoritmillä (listaus 7).

Listaus 7. Salasanan muodostus SHA-1-algoritmin avulla.

```
$salasana = mysql_real_escape_string($_POST["salasana"], $yhteys);  
$shalasana = sha1($salasana);
```

SHA-1-algoritmi muuttaa salasanan 40 merkin mittaiseksi tiivisteksi, eli käytännössä jonoksi sekalaisia merkkejä. Samasta salasanasta tulee SHA-1:n käsittelyssä aina sama tiiviste. Täten, kun salasana on luotu ja käyttäjä yrittää kirjautua sillä sisälle, verrataan hänen sillä hetkellä kirjoittamansa salasanan SHA-1-muotoa tietokannasta löytyvään merkkijonoon. Itse salasanaa ei siis käytetä tunnistautumiseen, vaan tämän tiivistettä. Tietokannassa oleva salasana-sarake tulee olla 40 merkin mittainen, jotta sinne pystytään tämä SHA-1-tiiviste tallentamaan.

Salasanojen säilöntään saadaan lisää turvallisuutta, kun käyttäjän kirjoittamaan salasanaan liitetään jokin monimutkainen merkkijono. Kun tämä merkkijonon lisääminen tapahtuu aina salasanaa luodessa ja myöhemmin kirjautumiseen liittyvässä tarkistuksessa, ei käyttäjän mahdollisesti hyvin yksinkertainen salasana ole uhka turvallisuudelle. Salasanojen SHA-1-tiivisteet selvittänyt hakkeri voisi verrat näitä löytämiään tiivisteitä helppojen ja useinkäytettyjen salasanoiden tiivistei-

siin. Kun salasanaa on pidennetty sivuston kehittäjän toimesta jollain merkkijonolla, ei tällainen vertailu onnistu. Itse käyttäjälle asia ei aiheuta mitään lisätyötä, vaan merkkijonon lisääminen hänen salasanaansa tapahtuu koodissa (listaus 8).

Listaus 8. Merkkijonon lisäys salasanaan.

```
$shalasana = sha1($salasana . "merkkijono");
```

Kun salasana on muutettu kirjautumisen yhteydessä tiivisteeksi, voidaan sitä ja käyttäjätunnusta verrata tietokannan *yllapito*-taulusta löytyviin tietoihin. Jos vastaavuutta ei löydy, antaa sivu virheilmoituksen. Mikäli taulusta löytyy samanlaisen tunnuksen ja salasanan sisältävä rivi, voi käyttäjän päästää etenemään varsinaiselle *yllapito*-sivulle. Ennen seuraavalle sivulle siirtymistä käytetään kuitenkin *session_start*-funktiota, jolla aloitetaan uusi istunto (listaus 9).

Listaus 9. Uuden istunnon aloittaminen.

```
session_start();  
$_SESSION["kayttaja_id"] = $rivi["id"];
```

Istunnon aloittaminen tarkoittaa istuntotiedoston luomista palvelimelle ja evästeen tallentamista käyttäjän tietokoneelle. *isset*-funktion avulla hyödynnettävä *\$_SESSION*-taulukko saa arvon *yllapito*-taulun *id*-sarakkeesta ja tallentuu palvelimen istuntotiedostoon. Tätä arvoa hyödynnetään käyttäjän tunnistamiseen kaikilla ylläpidon sivuilla. Evästeen avulla yhdistetään kyseinen käyttäjä, eli käyttäjän selain, istuntotiedostoon.

Eväste tuhoutuu, kun sen voimassaoloaika menee umpeen (oletuksena 24 minuuttia), kun selain suljetaan tai kun se tuhotaan kirjautumalla ulos. Uloskirjautuminen tarkoittaa käytännössä evästeen voimassaoloajan asettamista menneisyyteen, jolloin se tulkitsee olleensa käyttämättömänä yli sallitun aikarajan ja poistaa itsensä.

Listaus 10. Uloskirjautuminen.

```

session_start();
if (isset($_COOKIE[session_name()]))
    setcookie(session_name(), "", time()-50000, '/');
session_destroy();

```

Uloskirjautuminen sujuu siis siten, että istunto otetaan käyttöön ja etsitään tähän liittyvä eväste *isset*-funktiolla. Tämän jälkeen evästeen arvo tyhjäetään ja sen voimassaoloaika muutetaan menneisyyteen. Lauseessa oleva *'/'* tarkoittaa evästeen polkua, joka on tässä tapauksessa juuressa. Lopulta *session_destroy* poistaa istunnon palvelimelta. Uloskirjautumisen koodi on nähtävillä listauksessa 10.

4.4.4 Ylläpidon tunnistautuminen

Jokaisella ylläpidon sivulla suoritetaan aluksi kolme asiaa. Ensiksi kutsutaan *funktiot.php*-tiedostoa, sen jälkeen käytetään *session_start*-funktiota ja lopuksi kutsutaan *tarkista_kirjautuminen*-funktioita *funktiot.php*:sta. *session_start* jatkaa käynnissä olevaa istuntoa, jolloin eväste aloittaa sen voimassaoloajan laskemisen jälleen alusta. *tarkista_kirjautuminen* puolestaan tarkistaa, onko käyttäjä kirjautuneena sisälle. Tarkistaminen tehdään etsimällä *\$_SESSION*-taulukosta arvoa nimeltä *kayttaja_id*. Arvon löytyessä verrataan sitä *yllapito*-taulun *id*-sarakkeen sisältöön. Vastaavuuden löytyessä käyttäjän todetaan olevan kirjautuneena ja hänen annetaan jatkaa kyseessä olevalla sivulla. Jos ehto ei täyty, ohjataan käyttäjä automaattisesti kirjautumissivulle. *tarkista_kirjautuminen*-funktioon tutustutaan tarkemmin listauksessa 11.

Listaus 11. *funktiot.php*:ssa sijaitseva *tarkista_kirjautuminen*-funktio.

```
function tarkista_kirjautuminen()
{
    $yhteys = yhdistä_tietokantaan();
    if(isset($_SESSION["kayttaja_id"]))
    {
        $kayttaja_id = mysql_real_escape_string($_SESSION["kayttaja_id"], $yhteys);
        $tulokset = mysql_query("SELECT id, tunnus, tyyppi FROM yllapito
WHERE id = '$kayttaja_id'", $yhteys);
        if(!$tulokset)
            exit("Tietokantahaku ei onnistunut!");

        $rivi = mysql_fetch_assoc($tulokset);

        if($rivi["id"] == true)
        {return $rivi;}
        else
        {uudelleenohjaa("kirjautuminen.php");}
    }
    else
        uudelleenohjaa("kirjautuminen.php");
    exit;
}
```

Toinen tarkistettava asia on sisäänkirjautuneen käyttäjätilin tyyppi. Ainoastaan master-oikeudet omaava käyttäjä on oikeutettu muokkaamaan muita kuin ottelutietoja. Näin ollen jos admin-tason käyttäjä yrittää päästä osoiteriviä käyttämällä esimerkiksi muokkaamaan uutisia, niin hänen pääsynsä tulee evätä.

Ratkaisuni pääsyn rajoittamiseen käyttäjätyyppin perusteella on yksinkertainen. Sivun alussa tarkistetaan kaikkien ylläpidolle kuuluvien sivujen tapaan käyttäjän sisäänkirjautuminen, jonka aikana *tarkista_kirjautuminen*-funktion tulos sijoitetaan *\$kayttajatiedot*-muuttujaan. Tämän lisäksi sivut, joille admin-tyypin ylläpitäjällä ei ole oikeutta mennä, sisältävät koodissaan if-ehdon "*if(\$kayttajatiedot["tyyppi"] == 'Master')*". Jos ehto täyttyy, näytetään sivun sisältö. Jos käyttäjä on admin, ohjataan hänet suoraan takaisin ylläpidon aloitus sivulle.

4.4.5 Ylläpidon lomakkeet

Tietokannan tietojen lisäys, muuttaminen ja poisto tapahtuu lomakkeiden avulla. Uutisten, pelaajien ja joukkueiden kanssa tämä on aika yksinkertaista ja jokainen näistä toimii samalla periaatteella. Otan esimerkiksi uuden uutisen lisäämisen.

Uutta uutista luodessa täytettävissä on vain kaksi kenttää, joista toiseen kirjoitetaan otsikko ja toiseen uutisen varsinainen teksti. Kun lomakkeen *tallenna*-painiketta painetaan, suoritetaan annetuille tiedoille pienimuotoinen tarkistus (lista 12).

Listaus 12. Ote lomakkeelle kirjoitettujen tietojen virheentarkistuksesta.

```
$otsikon_pituus = mb_strlen($_POST["otsikko"]);
if($otsikon_pituus < 1)
$virheita := "Et voi jättää otsikko-kenttää tyhjäksi! ";
```

Tämä koodi tarkistaa kaikessa yksinkertaisuudessaan, että otsikko-kenttään on kirjoitettu jotain. *mb_strlen*-funktio siis laskee *otsikko*-arvon pituuden merkeissä ja tämän lukeman pituutta testataan *if*-lauseessa.

Listaus 13. Virhetilanteissa tulostetaan ongelmien kuvaukset.

```
if($virheita != "")
echo "<p class=\"virhe\"> ", $virheita, "</p>";
```

Jos virheitä havaitaan, silloin tallennus keskeytyy ja sivulle tulee lukemaan havaitun virheen kuvaus (lista 13). Kun annetut arvot läpäisevät virheentarkistuksen, käsitellään ne *mysql_real_escape_string*-funktioilla ja sijoitetaan muuttujiin. Näitä muuttujia voidaan sitten käyttää hyväksi SQL-lauseessa. Vaikka nyt tallennetaan haun sijaan, on toiminta käytännössä ihan samanlaista kuin aiemmin pelaajan tietoja hakiessa. Tällä kertaa muuttujia käytetään vain hyväksi *insert*-lauseessa (lista 14).

Listaus 14. Muuttujien käyttö tietokantaan tietoja lisätessä.

```
INSERT INTO uutiset (id, pvm, otsikko, teksti)
VALUES (null, now(), '$otsikko', '$teksti');
```

Ottelutietojen lisäys eroaa muista siinä suhteessa, että uusien tietojen luominen tapahtuu kahden lomakkeen avulla ja tietoa tallennetaan kolmeen tauluun. *Ottelu*-tauluun menevät perustiedot kuten ottelun ajankohta, *joukkue_ottelu*-tauluun lisä-

tään tiedot molempien peliin osallistuneiden joukkueiden suoriutumisesta ja *pelaaja_ottelu*-tauluun luodaan uudet rivit jokaisesta ottelussa pelanneesta pelaajasta. Kaikki nämä tiedot tulee olla yhdistettävissä toisiinsa. Tämän ongelman olen ratkaissut ottelutunnuksen luomisella.

Ottelutietojen lisäyksen ensimmäisellä lomakkeella määritellään *ottelu*- ja *joukkue_ottelu*-tauluihin menevät tiedot. Kun *tallenna*-painiketta painetaan, luodaan sivun koodissa ennen tietojen tallentamista ottelulle ottelutunnus (listaus 15). Jokaisen ottelun yksilöivä tunnus on esimerkiksi muotoa BERFOR-R1. Ensimmäiset kuusi kirjainta tarkoittavat osallistuneiden joukkueiden *joukkue*-taulusta löytyviä kolmen kirjaimen lyhenteitä. Väliviivan jälkeen seuraava R tarkoittaa runkosarjaa ja numero 1 kertoo, että kyseessä on joukkueiden ensimmäinen kohtaaminen runkosarjassa.

Listaus 15. Ottelutunnuksen luominen.

```
$tunnuso1 = mysql_query("select lyhenne from joukkue where
                        nimi = '$joukkue_koti'", $yhteys);
$tunnuso2 = mysql_query("select lyhenne from joukkue where
                        nimi = '$joukkue_vieras'", $yhteys);

if($vaihe == 'Runkosarja'){ $tunnuso3 = 'R'; }
else{ $tunnuso3 = 'P'; }

$tunnuso4 = mysql_query("select count(id) + 1 as maara from ottelu where
                        joukkue_koti = '$joukkue_koti' and joukkue_vieras = '$joukkue_vieras'
                        and vaihe = '$vaihe'
                        or joukkue_koti = '$joukkue_vieras' and joukkue_vieras = '$joukkue_koti'
                        and vaihe = '$vaihe'", $yhteys);

$rivi_ot1 = mysql_fetch_assoc($tunnuso1);
$rivi_ot2 = mysql_fetch_assoc($tunnuso2);
$rivi_ot4 = mysql_fetch_assoc($tunnuso4);

$otunnus = $rivi_ot1['lyhenne'].$rivi_ot2['lyhenne']."-".$tunnuso3.$rivi_ot4['maara'];
```

Luotu tunnus antaa jo itsessään paljon informaatiota ottelusta. Se kertoo, mikä joukkue on ollut kotijoukkue ja mikä joukkue on ollut vierasjoukkue. Se kertoo myös, onko kyseessä ollut pudotuspeli- vai runkosarjaottelu. Lopussa oleva numero yksilöi ottelun muista joukkueiden välisistä kohtaamisista ja kertoo samalla, kuinka monta ottelua näiden joukkueiden kesken on ollut tässä vaiheessa kautta (eli pudotuspeleissä tai runkosarjassa).

Ottelutunnus tallennetaan muiden tietojen kanssa *ottelu-* ja *joukkue_ottelu-*tauluihin. Samalla siirrytään ottelutietojen tallennuksessa toiselle lomakkeelle, jossa lisätään ottelun pelaajia koskevat tiedot. Tässä vaiheessa ottelutunnus on kulkeutunut osoiteriville, josta se lisätään jokaisen pelaajan tietojen kanssa *pelaaja_ottelu-*tauluun. Näin kaikki kolme taulua ovat yhteydessä toisiinsa ja ottelun tiedot saadaan tallennettua mahdollisimman laajasti.

4.5 Sivujen kuvaukset

Sivusto jakautuu kuuteen pääosastoon. Navigointipalkissa on linkit sivuille *uutiset*, *tulokset*, *tilastot*, *säännöt*, *tietoa* ja *ylläpito*. Näiden lisäksi sivustoon kuulua monia eri alasivuja. Seuraavaksi esittelen sivut yksi kerrallaan. Kuvat näistä sivuista ovat nähtävillä samassa järjestyksessä liitteenä (liitteet 2–20).

Uutiset-sivun sisältö perustuu tietokannan *uutiset*-tauluun. Taulun sisältö tuodaan näkyville muodossa päivämäärä, otsikko ja teksti. Jos tietokannassa on enemmän uutisia kuin kymmenen, jaetaan nämä usealle sivulle. *Uutiset* toimii sivuston etusivuna.

Tulokset-sivulla on liigan sarjataulukko sekä lista pelatuista otteluista. Sarjataulukko pohjautuu *joukkue_ottelu*-tauluun ja ottelutiedot *ottelu*-tauluun. Sarjataulukossa näkyvät joukkueen sijoitus, nimi, voitot, häviöt, tasapelit, pisteet, tehdyt maalit, päästetyt maalit ja maaliero. Taulukkoa voi lajitella näiden tietojen perusteella ja vertailla täten helposti joukkueiden suoritusta toisiinsa nähden eri kategorioissa. Joukkueen nimeä napsauttamalla pääsee *joukkue*-sivulle ja ottelulistasta jotain yksittäistä peliä napsauttamalla pääsee katsomaan tämän pelin tarkat tiedot *ottelu*-sivulle.

Tilastot-sivulla näkyy liigan pistepörssi, joka haetaan sivua ladatessa tietokannan *pelaaja_ottelu*-taulun tiedoista. Muodostuneen listauksen pystyy lajittelemaan pisteiden lisäksi myös maalien ja syöttöjen perusteella. Pelaajan nimeä napsauttamalla pääsee *pelaaja*-sivulle ja pelaajan joukkuetta napsauttamalla *joukkue*-sivulle. Sivulta löytyy myös hakutoiminto, jolla voi etsiä pelaajia nimen perusteella.

Joukkue-sivun muodostuminen riippuu osoiterivillä olevasta arvosta. Arvo muodostuu sivulle tuoneen linkin perusteella, eli jos napsauttaa esimerkiksi *tulokset*-sivun sarjataulukosta joukkueen X nimeä, tulee näkyville *joukkue*-sivu arvolla X, eli joukkueen X tiedot. Näihin tietoihin kuuluu *joukkue*-taulusta esille tulevat perustiedot, *pelaaja_ottelu*-taulun perusteella tuotava joukkueen sisäinen pistepörssi (jossa hakutuloksien ehdoksi laitetaan se, että pelaajan täytyy pelata joukkueessa X) ja *ottelu*-taulusta esille tuotavat joukkueen menneet ottelut. Pistepörssissä näkyviä pelaajien nimiä napsauttamalla pääsee *pelaaja*-sivulle ja ottelutietoja napsauttamalla *ottelu*-sivulle.

Pelaaja-sivun näyttämä sisältö riippuu *joukkue*-sivun tyylistä osoiterivillä olevasta arvosta, eli tässä tapauksessa pelaajan nimestä. Esille tulevat pelaajan perustiedot *pelaaja*-taulusta, pelaajan kausikohtaiset tilastot *pelaaja_ottelu*-taulusta ja listaus pelaajan pelaamista otteluista *ottelu*-taulusta. Sivulta pääsee siirtymään joko joukkueen nimeä napsauttamalla *joukkue*-sivulle tai ottelulistasta *ottelu*-sivulle.

Ottelu-sivu hakee osoiteriviltä ottelutunnusta. Ottelutunnuksen perusteella se näyttää tiedot ottelusta (*ottelu*-taulu) ja siinä pelanneista pelaajista, sekä heidän tilastoistaan (*pelaaja_ottelu*-taulu). Sivulta voi napsauttaa pelaajien tai joukkueiden nimiä, jolloin pääsee katsomaan näiden sivuja.

Säännöt ja *tietoa* ovat sivuston yksinkertaisimmat sivut toteutuksensa kannalta. Niissä ei käytetä ollenkaan tietokantaa, vaan sivuilla näkyvä teksti on kirjoitettu suoraan PHP-tiedostoihin. Tämä siksi, että tekstiä ei ole tarvetta muokata. Säännöt pysyvät kaudesta toiseen samoina, eikä tietosivulla näkyvä teksti liigan taustoista tarvitse päivitystä. Mikäli jotain pientä muutosta kuitenkin joskus ilmenisi, niin tämän muutoksen pystyy tekemään sivuston varsinainen ylläpitäjä helposti php-tiedostoja muuttamalla.

Ylläpito-sivulla on mahdollisuus sisäänkirjautumiseen, joka tapahtuu ylläpidon jäsenille jaetuilla tunnuksilla ja salasanoilla. Näitä tunnuksia säilytetään *yllapito*-taulussa. Ylläpitäjä voi olla käyttäjätyypiltään joko master tai admin. Jälkimmäinen pystyy vaikuttamaan ainoastaan ottelutietoihin, master-luokan käyttäjän kyetessä muokkaamaan myös uutisia sekä pelaaja- ja joukkueetietoja. Master-

tunnukset myönnetäänkin vain muutamalle liigan johtohenkilölle, kun taas loput vastuuhenkilöistä tarvitsevat ainoastaan oikeudet lisätä tietokantaan uusia otteluita.

Sisäänkirjautumisen jälkeen admin-tason käyttäjän eteen avautuu siis ainoastaan ottelut-linkki ja mahdollisuus kirjautua ulos. Ottelut-linkkiä napsauttamalla saa nähtäville listauksen kaikista tietokantaan lisätyistä otteluista ja linkin, josta pääsee luomaan uuden ottelun. Listattuja otteluita voi myös napsauttaa, jolloin pääsee muokkaamaan näiden tietoja. Listaukseen kuuluu jokaisen ottelun vieressä oleva POISTA-linkki, jonka kautta saa poistettua väärin lisätyt ottelutiedot. Tätä toimenpidettä suorittaessa kysytään vielä käyttäjän varmistus, jotta välttytään virheil-
tä.

Uuden ottelun luonti tapahtuu kahden hyvin selkeän lomakkeen kautta. Ensiksi täytetään ottelun perustiedot joukkueiden, näiden tekemien maalien ja ottelun vaiheen (esimerkiksi runkosarjaottelu tai finaaliottelu) osalta. Lisäksi ottelusta voi kirjoittaa raportin, mutta se ei ole pakollista. Tiedot tallennetaan *ottelu* ja *joukkue_ottelu* -tauluihin. Ensimmäisen lomakkeen tietojen tallennuksen jälkeen päästään kirjaamaan ottelussa mukana olleet pelaajat ja näiden pisteet. Nämä tiedot tallennetaan *pelaaja_ottelu*-tauluun. Vaihe, joukkueet ja pelaajat täytetään alasvetovalikkojen avulla, jolloin virheet minimoidaan ja asiat tehdään mahdollisimman helpoksi. Ottelutietojen muokkauksessa käytetään näitä samoja sivuja sillä erotuksella, että lomakkeet on täytetty valmiiksi aiemmin annetuilla tiedoilla.

Master-tason käyttäjä saa mahdollisuuden lisätä, muokata ja poistaa otteluiden lisäksi myös pelaajia, joukkueita ja uutisia. Kaikkien näiden kanssa toimiminen tapahtuu samaan tyyliin kuin otteluidenkin. Tietokannan vaatimat tiedot täytetään selkeisiin lomakkeisiin ja napsautetaan *tallenna*-painiketta. Mitään ohjeita ei lomakkeisiin tai muihin sivun toimintoihin kuulu. Toiminnot selittävät itse itsensä ja kun käyttäjät ovat Internetissä pelattavan moninpelin pelaajia, he kyllä osaavat näin yksinkertaista sovellusta käyttää.

5 LOPPUTULOS JA JOHTOPÄÄTÖKSET

Aloittaessani opinnäytetyön tekemisen minulla oli kokemusta lähinnä yksinkertaisten verkkosivujen luomisesta ja tietokantojen kanssa olin työskennellyt melkeinpä ainoastaan Microsoft Accessin avulla. Nyt, työn valmistuttua, olen onnistunut luomaan suunnitelmieni mukaisen tietokantaa käyttävän verkkosivuston. Olen siis oppinut ne asiat, jotka halusinkin oppia, ja saanut aikaiseksi tavoitteena olleen lopputuloksen. Näin ollen pidän opinnäytetyötäni onnistuneena.

Koin oman aiheen valitsemisen mielekkääksi, sillä näin olin vastuussa ainoastaan itselleni. Ennen työn aloitusta olin epäileväinen omiin kykyihini toteuttaa jonkun toimeksiantajan tarvitsemaa vastaavanlaista tulosta. Asiaan perehdyttyäni sain kuitenkin lopulta huomata, että pienen opettelun jälkeen taitoa karttui ja sain aikaiseksi hyvää jälkeä.

Sivuston vaatimaan sisältöön sain mallia vastaavien Kiekko.tk-liigojen sivuilta, sekä myös ihan oikeiden jääkiekkosarjojen sivuja tutkimalla. Sarjataulukko ja pistepörssi tuodaan tällaisilla sivuilla parhaiten esille, koska ne ovat sivujen seurautuinta sisältöä. Myös ottelu- ja pelaajakohtaiset tulokset ja tilastot ovat ahkerassa käytössä. Nämä tiedot ovat minun sivustollani hyvin esillä ja luotankin vakaasti siihen, että käyttäjät löytävät haluamansa.

Lopputuloksena oleva sivusto on käytettävyydeltään selkeä ja sisältää tarvittavat toiminnot. Erityisen tyytyväinen olen ylläpito-osiossa tapahtuvaan ottelutietojen lisäykseen, joka kahden erittäin yksinkertaisen lomakkeen avulla lisää tietoa kolmeen eri tauluun pelatusta ottelusta ja siihen osallistuneiden joukkueiden ja pelaajien suoriutumisesta. Tietojen jakaminen eri tauluihin kyseisellä tyylillä ja näiden tietojen yhdistäminen toisiinsa lomakkeen täytön aikana luotavalla ottelutunnuksella oli hyvä ja ennen kaikkea toimiva oivallus.

Yllämainitun ottelutietojen säilönnän keksiminen oli monimutkaista ja haastavaa. Sivuston koodin toteutus aiheutti myös ajoittaista turhautumista ongelmakohtien kanssa. Lopulta kuitenkin löysin aina itse ratkaisun ongelmiin ja yleensä nämä

ongelmat olivat vain jotain pienen pieniä yksityiskohtia, jotka olivat jääneet syys-tä tai toisesta huomioimatta.

Jälkeenpäin ajateltuna sivuston toteutuksessa ei ole mitään, jonka tekisin toisin. Toteutusratkaisut voivat olla paikoittain aika yksinkertaisia asiaan perehtyneem-män silmään, mutta ne ajavat asiansa. Asettamani tavoitteet tulivat täytetyksi ja sivusto on täysin toimiva. Itse opinnäytetyöprosessissa työskentely olisi voinut olla ajoittain tehokkaampaa, jotta prosessin kokonaiskesto olisi ollut suppeampi.

Tarkoituksena ei ole ottaa tällä hetkellä sivustoa oikeaan käyttöön, mutta se olisi täysin mahdollista. Ainoina toimenpiteinä pitäisi hankkia liigalle oma domain, tietokannan käytön mahdollistava kotisivutila ja pari aiheelle omistautunutta hen-kilöä auttamaan liigan pyörittämisessä. Innostuksen löytyessä sivuston käyttöön-otto voi siis edelleen tapahtua.

Tärkein tästä opinnäytetyöstä seurannut asia on kuitenkin se, että nyt minulla on tiedot ja taidot luoda tietokantaa käyttäviä verkkosivustoja. Tämä on korvaamaton apu niin työelämässä kuin tiettyjen harrastuksien kanssa toimimisessakin.

LÄHTEET

Heinisuo, Rami & Rauta, Ilkka (2007). 4. painos. PHP ja MySQL Tietokantapohjaiset verkkopalvelut. Helsinki. Talentum.

Kolehmainen, Kauko (2006). 1. painos. PHP & MySQL - Teoriasta käytäntöön. Helsinki. Readme.fi.

Korpela, Jukka (2003). 1. painos. CSS-tyylit. Jyväskylä. Docendo.

Krug, Steve (2006). 2. painos. Älä pakota minua ajattelemaan. Helsinki. Readme.fi.

Pynnönen, Jouko & Pynnönen, Mikko (2011). [viitattu 13.6.2011] Saatavilla Internetissä: <URL: <http://kiekko.tk/about.cws>>.

w3schools.com (2011). [viitattu 25.10.2011] Saatavilla Internetissä: <URL: <http://www.w3schools.com/html/>>.

LIITELUETTELO

LIITE 1. Tietokannan ER-kaavio

LIITE 2. EHKL-sivuston uutiset-sivu

LIITE 3. EHKL-sivuston tulokset-sivu

LIITE 4. EHKL-sivuston tilastot-sivu

LIITE 5. EHKL-sivuston joukkue-sivu

LIITE 6. EHKL-sivuston pelaaja-sivu

LIITE 7. EHKL-sivuston ottelu-sivu

LIITE 8. EHKL-sivuston säännöt-sivu

LIITE 9. EHKL-sivuston tietoa-sivu

LIITE 10. EHKL-sivuston kirjautuminen-sivu

LIITE 11. EHKL-sivuston ylläpito-sivu

LIITE 12. EHKL-sivuston ylläpito: otteluiden selaus -sivu

LIITE 13. EHKL-sivuston ylläpito: otteluiden muokkaus 1 -sivu

LIITE 14. EHKL-sivuston ylläpito: otteluiden muokkaus 2 -sivu

LIITE 15. EHKL-sivuston ylläpito: pelaajien selaus -sivu

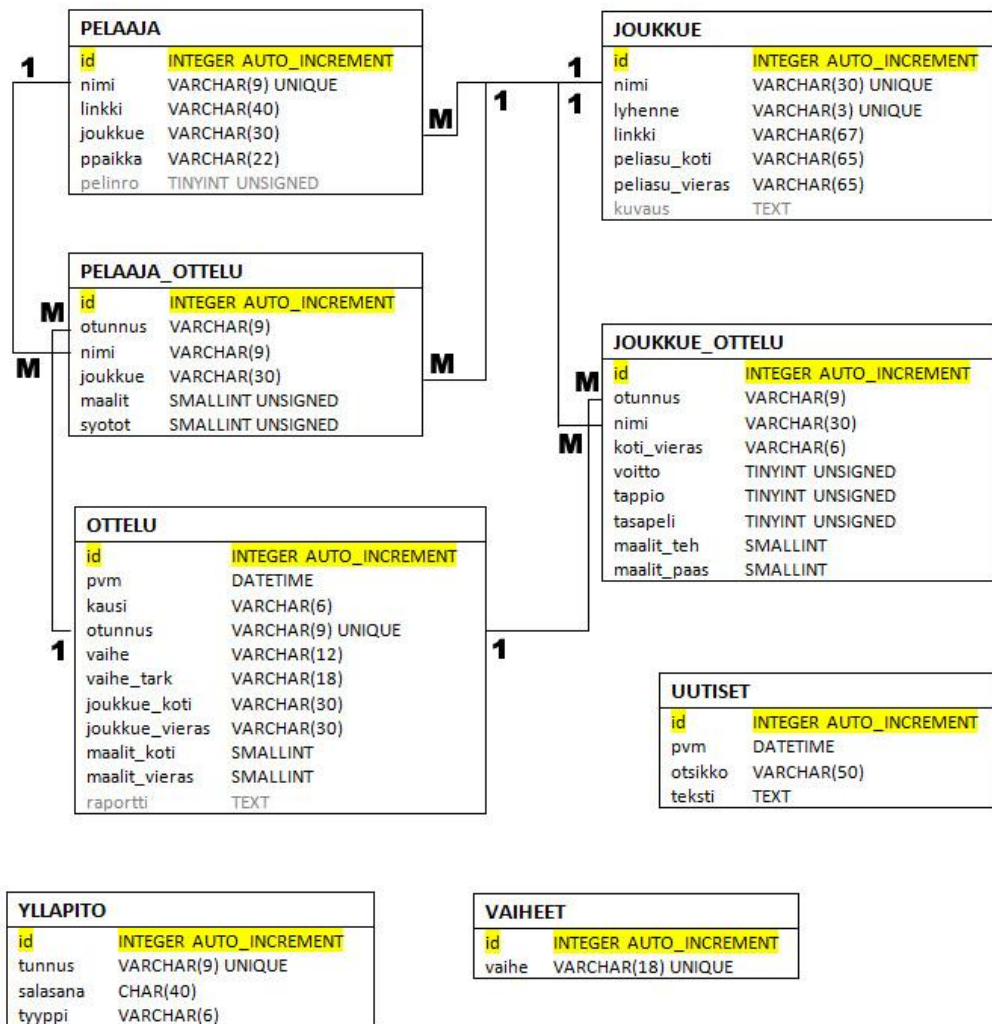
LIITE 16. EHKL-sivuston ylläpito: pelaajien muokkaus -sivu

LIITE 17. EHKL-sivuston ylläpito: joukkueiden selaus -sivu

LIITE 18. EHKL-sivuston ylläpito: joukkueiden muokkaus -sivu

LIITE 19. EHKL-sivuston ylläpito: uutisten selaus -sivu

LIITE 20. EHKL-sivuston ylläpito: uutisten muokkaus -sivu





UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Uutiset

1 2

24.10.2011 | Runkosarjan päätös lähenee

Muistutuksena joukkueille, joilla on vielä pelejä pelaamatta: Otteluita saa pelata tämän viikon loppuun saakka. Pelien tulee siis olla pelattuna sunnuntaina 30.11 klo 23:59 mennessä.

Mikäli joku joukkue ei saa pelattua tarvittavia pelejä määräaikaan mennessä, tuomitaan tälle näistä otteluista tappiot. Lisäksi syyllinen osapuoli otteluiden pelaamattomuuteen tullaan tutkimaan ja seurauksena voi olla pahimmassa tapauksessa liigasta sulkeminen.

11.9.2011 | Pelikielto: yberpeelo

Pelaaja nimeltä yberpeelo tuomitaan määrittelemättömän pituiseen pelikieltoon. Täten hän ei saa osallistua EHKL:n toimintaan millään tavoin missään joukkueessa.

Syynä rangaistukseen ovat pelaajan toistuneet sääntörikkomukset.

10.9.2011 | Kausi 2011/1 käynnistynyt

EHKL:n ensimmäinen kausi on alkanut! Jokainen osallistuva joukkue tulee pelaamaan 28 ottelua runkosarjaa, jonka jälkeen kahdeksan parasta etenee pudotuspeleihin.

Runkosarjan ottelut sisältävät kaksi peliä jokaista joukkuetta vastaan. Toinen otteluista pelataan kotipelinä ja toinen vieraisissa. Pelien järjestyksellä ja niiden pelaamisen ajankohdalla ei ole mitään väliä, kunhan ne saadaan pelattua ennen aikarajan loppua.

Pudotuspelit pelataan paras viidestä -systeemillä, eli kolmella voitolla jatkoon. Parit määräytyvät jokaisella kierroksella runkosarjasijoitusten perusteella, eli kaikissa vaiheissa jatkossa olevista joukkueista parhaiten sijoittunut pelaa huonointen sijoittunutta vastaan, toiseksi paras toiseksi huonointa jne. Välieränsä hävinneet joukkueet pelaavat pronssipelin (eli yhdellä voitolla ratkeavan).



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Sarjataulukko

Kausi: 2011/1

# Joukkue	Ottelut	Voitot	Tappiot	Tasapelit	Pisteet	TM	PM	Maaliero
1 Berliinin Munkki	26	13	6	7	33	90	79	11
2 Hampaattomat Kanadalaiset	24	13	7	4	30	96	61	35
3 Isän Maan Toivot	23	10	5	8	28	70	59	11
4 Fortune Club	21	10	4	7	27	73	56	17
5 Norjan Kalastelijat	21	11	6	4	26	80	51	29
6 Washingtonin Isot Kirjaimet	21	10	5	6	26	75	51	24
7 Venäjän Mafia	22	11	7	4	26	76	60	16
8 Työttömät Hitsarit	23	8	6	9	25	62	56	6
9 Jurvan Juntit	24	8	11	5	21	55	70	-15
10 Stadin Ilmaveivit	21	7	8	6	20	54	53	1
11 Laihian Alisuorittajat	23	6	10	7	19	53	63	-10
12 Team53	22	6	10	6	18	49	60	-11
13 HT Hokkarit	24	7	14	3	17	48	84	-36
14 Hollywoodin Filmaajat	21	5	11	5	15	45	56	-11
15 Sneezing Pandas	28	5	20	3	13	43	110	-67

Ottelut

	1 2 3 4 5 6
24.10.2011 Laihian Alisuorittajat - HT Hokkarit	2 - 0
24.10.2011 Hampaattomat Kanadalaiset - Venäjän Mafia	5 - 4
24.10.2011 Washingtonin Isot Kirjaimet - Työttömät Hitsarit	6 - 6
22.10.2011 Washingtonin Isot Kirjaimet - Team53	0 - 0
22.10.2011 Venäjän Mafia - Team53	2 - 1
22.10.2011 Fortune Club - Team53	3 - 2
22.10.2011 Fortune Club - HT Hokkarit	5 - 5
22.10.2011 Fortune Club - Hampaattomat Kanadalaiset	3 - 1
22.10.2011 Berliinin Munkki - Työttömät Hitsarit	2 - 2
20.10.2011 Työttömät Hitsarit - Berliinin Munkki	0 - 4
20.10.2011 Isän Maan Toivot - Berliinin Munkki	3 - 3



UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Pistepörssi

Kausi: 2011/1

Vaihe: Runkosarja

 Hae pelaaja

#	Pelaaja	Joukkue	Ottelut	Maalit	Syötöt	Pisteet
1	Ole-Olof	Norjan Kalastelijat	21	23	36	59
2	es-espen	Norjan Kalastelijat	21	37	21	58
3	Flööri	Hampaattomat Kanadalaiset	24	33	25	58
4	Hietala	Berliinin Munkki	26	39	16	55
5	SirMazku	Berliinin Munkki	26	20	35	55
6	Oveski	Washingtonin Isot Kirjaimet	21	39	15	54
7	Cosby	Hampaattomat Kanadalaiset	24	22	30	52
8	Laajamies	Washingtonin Isot Kirjaimet	21	13	39	52
9	k-apina	Fortune Club	21	22	28	50
10	Jessica	Berliinin Munkki	26	26	23	49
11	Timbero	Fortune Club	21	22	26	48
12	Fronger	Hampaattomat Kanadalaiset	24	19	29	48
13	aatuh	Isän Maan Toivot	23	28	18	46
14	saportsky	Työttömät Hitsarit	23	37	8	45
15	vellu44	Isän Maan Toivot	23	20	25	45
16	Maario	Hampaattomat Kanadalaiset	7	19	26	45
17	tilperok	Fortune Club	21	29	13	42
18	Bjorn	Norjan Kalastelijat	21	23	19	42
19	Mongoilny	Venäjän Mafia	22	22	19	41
20	ArmyJoe	Työttömät Hitsarit	23	25	15	40
21	valametti	Jurvan Juntit	24	23	16	39
22	konstar	Venäjän Mafia	20	19	19	38
23	Metsänen	Stadin Ilmaveivit	19	13	25	38
24	Seimi	Washingtonin Isot Kirjaimet	21	22	15	37
25	Crabitar	Hollywoodin Filmaajat	21	20	17	37
26	Verhorov	Venäjän Mafia	18	6	30	36
27	Purenta	Venäjän Mafia	12	28	5	33
28	Bauerauer	HT Hokkarit	24	20	13	33



EHKL

ERINOMAINEN HIETALAN KIEKKO-TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Berliinin Munkki

<http://kiekko.tk/teams/team.cws?team=Berliinin+Munkki>

Peliasu kotona: Valkoinen - Sininen - Valkoinen (8)

Peliasu vieraissa: Sininen - Valkoinen - Sininen (8)

Joukkueen kuvaus:

2005 vuoden lopulla perustettu joukkue löi pillit pussiin 2006 vuoden kesällä. Joukkueen uusi tuleminen alkoi vuoden 2011 alussa.

Pistepörssi

Kausi: 2011/1

Vaihe: Runkosarja

# Pelaaja	Ottelut	Maalit	Syötöt	Pisteet
1 Hietala	26	39	16	55
2 SirMazku	26	20	35	55
3 Jessica	26	26	23	49

Ottelut

Kausi: 2011/1

22.10.2011	Berliinin Munkki - Työttömät Hitsarit	2 - 2
20.10.2011	Työttömät Hitsarit - Berliinin Munkki	0 - 4
20.10.2011	Isän Maan Toivot - Berliinin Munkki	3 - 3
20.10.2011	Berliinin Munkki - HT Hokkarit	4 - 1
9.10.2011	Berliinin Munkki - Team53	6 - 4
9.10.2011	Team53 - Berliinin Munkki	4 - 4
9.10.2011	Venäjän Mafia - Berliinin Munkki	5 - 5
9.10.2011	Berliinin Munkki - Venäjän Mafia	3 - 2
9.10.2011	Washingtonin Isot Kirjaimet - Berliinin Munkki	3 - 3
5.10.2011	Norjan Kalastelijat - Berliinin Munkki	4 - 6
5.10.2011	Berliinin Munkki - Washingtonin Isot Kirjaimet	0 - 8
5.10.2011	Berliinin Munkki - Isän Maan Toivot	2 - 6



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTiset :: Tulokset :: Tilastot :: SÄÄNNÖT :: Tietoa :: Ylläpito

#21 Hietala

<http://kiekko.tk/user.cws?name=Hietala>

Joukkue: Berliinin Munkki

Pelipaikka: Hyökkääjä

Runkosarjatilastot

Kausi	Joukkue	Ottelut	Maalit	Syötöt	Pisteet
2011/1	Berliinin Munkki	26	39	16	55
	Yhteensä	26	39	16	55

Ottelut

Kausi: 2011/1

22.10.2011	Berliinin Munkki - Työttömät Hitsarit	2 - 2 (0+0)
20.10.2011	Työttömät Hitsarit - Berliinin Munkki	0 - 4 (0+2)
20.10.2011	Isän Maan Toivot - Berliinin Munkki	3 - 3 (1+0)
20.10.2011	Berliinin Munkki - HT Hokkarit	4 - 1 (2+0)
9.10.2011	Berliinin Munkki - Team53	6 - 4 (3+0)
9.10.2011	Team53 - Berliinin Munkki	4 - 4 (2+0)
9.10.2011	Venäjän Mafia - Berliinin Munkki	5 - 5 (1+1)
9.10.2011	Berliinin Munkki - Venäjän Mafia	3 - 2 (2+0)
9.10.2011	Washingtonin Isot Kirjaimet - Berliinin Munkki	3 - 3 (1+0)
5.10.2011	Norjan Kalastelijat - Berliinin Munkki	4 - 6 (2+3)
5.10.2011	Berliinin Munkki - Washingtonin Isot Kirjaimet	0 - 8 (0+0)
5.10.2011	Berliinin Munkki - Isän Maan Toivot	2 - 6 (1+0)
5.10.2011	Berliinin Munkki - Laihian Alisuorittajat	4 - 3 (4+0)
1.10.2011	Laihian Alisuorittajat - Berliinin Munkki	4 - 5 (0+0)
1.10.2011	HT Hokkarit - Berliinin Munkki	1 - 2 (2+0)
27.9.2011	Berliinin Munkki - Jurvan Juntit	5 - 0 (2+1)
27.9.2011	Jurvan Juntit - Berliinin Munkki	4 - 2 (1+1)
25.9.2011	Berliinin Munkki - Norjan Kalastelijat	2 - 5 (1+1)

**EHL**

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

[UUTISET](#) :: [TULOKSET](#) :: [TILASTOT](#) :: [SÄÄNNÖT](#) :: [TIETOA](#) :: [YLLÄPITO](#)**Berliinin Munkki - Fortune Club 5 - 3**

Pelattu: 10.9.2011

Kausi: 2011/1

Vaihe: Runkosarjaottelu

[Berliinin Munkki](#)[Hietala](#) 5+0=5[SirMazku](#) 0+5=5[Jessica](#) 0+2=2[Fortune Club](#)[Timbero](#) 1+2=3[k-apina](#) 1+1=2[tilperok](#) 1+0=1

Raportti:

Kauden ensimmäinen ottelu tarjosi valtaisa Kiekko-viihdettä.

**EHKL**

ERINOMAINEN HIIETALAN KIEKKO.TK-LIIGA

UUTISSET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Säännöt

Kauden eteneminen

Runkosarjassa joukkue pelaa kaksi kertaa jokaista vastaan. Toinen peleistä on kotipeli ja toinen vieraspelejä. Runkosarjan jälkeen kahdeksan parhaiten sijoittunutta joukkuetta pelaavat pareittain pudotuspelit paras viidestä -systeemillä, eli kolmella voitolla jatkoon. Sarjojen voittajat etenevät seuraavalle kierrokselle ja häviäjät putoavat. Parit määräytyvät jokaisella kierroksella runkosarjasijoitusten perusteella, eli kussakin vaiheessa jatkossa olevista joukkueista parhaiten sijoittunut pelaa huonointen sijoittunutta vastaan, toiseksi paras toiseksi huonointa jne. Parempi joukkue saa kotiedun, eli pääsee aloittamaan sarjan kotipelillä. Tämän jälkeen kotipeleissä vuorotellaan, joten täyden viiden pelin sarja menisi paremmin sijoittuneen joukkueen kannalta: Koti, vieras, koti, vieras ja koti. Välieränsä hävinneet joukkueet pelaavat pronssipelin (eli yhdellä voitolla ratkeavan).

Ottelut

Peleissä käytetään 3vs3 pelimuotoa. Joukkueen kokoonpanossa saa olla yhdessä ottelussa kuusi pelaajaa. Ottelussa pelataan kolme erää, joiden pituus on viisi minuuttia. Jos ottelu on pudotuspeleissä varsinaisen peliajan jälkeen tasan, niin pelataan äkkikuolema-jatkoajoja (5 min) siihen asti, että ratkaisu tulee.

Jokaisessa ottelussa tulee olla tuomari, joka kirjaa ylös maalit ja pisteet, sekä valvoo ettei sääntöjä rikota. Ottelu ei ole virallinen, jos pelissä ei ole tuomaria. Maaleja ei saa hylätä.

Jos ottelusta putoaa kenttäpelaaja, laitetaan pitkää SEURAAVAN KATKON JÄLKEEN maksimissaan minuutin ajan. Ottelu jatkuu siis normaalisti seuraavaan katkoon asti ja mahdolliset tällä välillä syntyneet maalit hyväksytään. Tämä siksi, että välttyään "sillä lagas" -syytelyltä maalin synnyttyä. Jos pudonnut pelaaja ei tule pelikellon näyttämässä minuutissa takaisin, ottelua jatketaan ilman häntä. Pelaajan palatessa / lajin loputtua minuutin sisällä, luovuttaa kyseisen pelaajan joukkue kiekon vastustajalle ja peli jatkuu. Jos sama pelaaja tipahtaa toisen kerran pelin saman aikana, ei tätä noteerata, vaan peli jatkuu normaalisti. Jos koko Kiekko.tk kaatuu kesken ottelun, peli pelataan myöhemmin uudestaan.

Publicilla puhuminen on kiellettyä ottelun aikana. Ainoastaan tuomareilla on tähän lupa. Tämä siksi että tuomarit ehtisivät ottaa maalintekijät ja -syöttäjät ylös tilastoja varten, ja että pelaajat eivät häiriinny. Teamchatissa puhuminen on sallittua katsojilta, mutta mikäli ottelussa on paljon väkeä, katsomo saatetaan hiljentää mute-komennolla. Pelaajat saavat teamchatissa puhua vapaasti.

Muuta

Huoneen nimeksi pitää laittaa: EHK1 (joukkue-joukkue) salasana ilmoitetaan pelaajille



[UUTISET](#) :: [TULOKSET](#) :: [TILASTOT](#) :: [SÄÄNNÖT](#) :: [TIETOA](#) :: [YLLÄPITO](#)

Tietoa

EHKL on internetissä pelattavan Kiekko.tk -moninpelin käyttäjille tarkoitettu liiga. Joukkueita liigaan mahtuu kerrallaan 15 kappaletta. Joukkuevalinnat tehdään karsintojen perusteella.

EHKL on nimensä mukaisesti Hietala-nimimerkillä Kiekko.tk:ssa tunnetun pelaajan perustama. Perustaminen tapahtui osana Hietalan ammattikorkeakouluun vuonna 2011 tehtyä opinnäytetyötä, jonka aiheena oli tietokantapohjaisen verkkosivun toteuttaminen.




[UUTISET](#) :: [TULOKSET](#) :: [TILASTOT](#) :: [SÄÄNNÖT](#) :: [TIETOA](#) :: [YLLÄPITO](#)

Ylläpito

Tämä sivu koskee ainoastaan EHKL:n ylläpitoa.

Käyttäjätunnus:

Salasana:



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA


UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO


Tervetuloa ylläpitoon Hietala


Käyttäjätilyi tyyppi on: Master


Olet oikeutettu tekemään muutoksia seuraaviin osioihin:

[Kirjautu ulos](#)

[Ottelut](#)

[Pelaajat](#)

[Joukkueet](#)

[Uutiset](#)



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO


Ylläpito | ottelut

Muokkaa otteluiden tietoja tai [luo uusi ottelu](#).

[Takaisin](#)

1 2 3 4 5 6

24.10.2011| [Laihian Alisuorittajat - HT Hokkarit](#) 2 - 0 [POISTA](#)
 24.10.2011| [Hampaattomat Kanadalaiset - Venäjän Mafia](#) 5 - 4 [POISTA](#)
 24.10.2011| [Washingtonin Isot Kirjaimet - Työttömät Hitsarit](#) 6 - 6 [POISTA](#)
 22.10.2011| [Washingtonin Isot Kirjaimet - Team53](#) 0 - 0 [POISTA](#)
 22.10.2011| [Venäjän Mafia - Team53](#) 2 - 1 [POISTA](#)
 22.10.2011| [Fortune Club - Team53](#) 3 - 2 [POISTA](#)
 22.10.2011| [Fortune Club - HT Hokkarit](#) 5 - 5 [POISTA](#)
 22.10.2011| [Fortune Club - Hampaattomat Kanadalaiset](#) 3 - 1 [POISTA](#)
 22.10.2011| [Berliinin Munkki - Työttömät Hitsarit](#) 2 - 2 [POISTA](#)
 20.10.2011| [Työttömät Hitsarit - Berliinin Munkki](#) 0 - 4 [POISTA](#)
 20.10.2011| [Isän Maan Toivot - Berliinin Munkki](#) 3 - 3 [POISTA](#)
 20.10.2011| [Berliinin Munkki - HT Hokkarit](#) 4 - 1 [POISTA](#)
 20.10.2011| [Washingtonin Isot Kirjaimet - Sneezing Pandas](#) 7 - 3 [POISTA](#)
 20.10.2011| [Washingtonin Isot Kirjaimet - Laihian Alisuorittajat](#) 3 - 3 [POISTA](#)
 20.10.2011| [Washingtonin Isot Kirjaimet - Jurvan Juntit](#) 2 - 0 [POISTA](#)
 19.10.2011| [Washingtonin Isot Kirjaimet - Isän Maan Toivot](#) 6 - 0 [POISTA](#)
 19.10.2011| [Washingtonin Isot Kirjaimet - HT Hokkarit](#) 4 - 0 [POISTA](#)
 19.10.2011| [Washingtonin Isot Kirjaimet - Hollywoodin Filmaajat](#) 4 - 4 [POISTA](#)
 19.10.2011| [Washingtonin Isot Kirjaimet - Hampaattomat Kanadalaiset](#) 1 - 6 [POISTA](#)
 19.10.2011| [Washingtonin Isot Kirjaimet - Fortune Club](#) 2 - 5 [POISTA](#)
 19.10.2011| [Venäjän Mafia - Työttömät Hitsarit](#) 3 - 0 [POISTA](#)
 19.10.2011| [Venäjän Mafia - Stadin Ilmaveivit](#) 2 - 2 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Sneezing Pandas](#) 6 - 2 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Laihian Alisuorittajat](#) 1 - 0 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Jurvan Juntit](#) 0 - 0 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Isän Maan Toivot](#) 3 - 9 [POISTA](#)
 17.10.2011| [Venäjän Mafia - HT Hokkarit](#) 2 - 3 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Hampaattomat Kanadalaiset](#) 6 - 5 [POISTA](#)
 17.10.2011| [Venäjän Mafia - Fortune Club](#) 4 - 4 [POISTA](#)
 17.10.2011| [Työttömät Hitsarit - Washingtonin Isot Kirjaimet](#) 5 - 1 [POISTA](#)



EHKL
ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTiset :: Tulokset :: Tilastot :: Säännöt :: Tietoa :: Ylläpito

Ylläpito | ottelut

Luo tai muokkaa otteluita. Huomaa, että ottelun joukkueita tai pudotuspeli / runkosarja -määrittelyä ei voi muuttaa jälkeenpäin!

Takaisin

Vaihe
▼

Kotijoukkue
▼

Vierasjoukkue
▼

Maalit

Maalit

Raportti

Tallenna



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

[UUTISET](#) :: [TULOKSET](#) :: [TILASTOT](#) :: [SÄÄNNÖT](#) :: [TIETOA](#) :: [YLLÄPITO](#)

Ylläpito | ottelut

Luo tai muokkaa pelaajien ottelukohtaisia tilastoja.

Takaisin

Berliinin Munkki

[illegible]

Fortune Club

[illegible]

Tallenna

**EHKL**


ERINOMAINEN HIIETALAN KIEKKO.TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Ylläpito | pelaajatMuokkaa pelaajien tietoja tai [luo uusi pelaaja](#).[Takaisin](#)

1 2

[aatuh](#) | Isän Maan Toivot | [POISTA](#)
[AMP](#) | Team53 | [POISTA](#)
[ArmyJoe](#) | Työttömät Hitsarit | [POISTA](#)
[Avecry](#) | Hollywoodin Filmaajat | [POISTA](#)
[Bauerauer](#) | HT Hokkarit | [POISTA](#)
[Bjorn](#) | Norjan Kalastelijat | [POISTA](#)
[Cosby](#) | Hampaattomat Kanadalaiset | [POISTA](#)
[Crabitar](#) | Hollywoodin Filmaajat | [POISTA](#)
[es-espen](#) | Norjan Kalastelijat | [POISTA](#)
[Flööri](#) | Hampaattomat Kanadalaiset | [POISTA](#)
[Fronger](#) | Hampaattomat Kanadalaiset | [POISTA](#)
[Grunlund](#) | Stadin Ilmaveivit | [POISTA](#)
[Hietala](#) | Berlinin Munkki | [POISTA](#)
[hitsipete](#) | Työttömät Hitsarit | [POISTA](#)
[Jessica](#) | Berlinin Munkki | [POISTA](#)
[k-apina](#) | Fortune Club | [POISTA](#)
[Kohta](#) | Stadin Ilmaveivit | [POISTA](#)
[konstar](#) | Venäjän Mafia | [POISTA](#)
[Kyntäjä](#) | Jurvan Juntit | [POISTA](#)
[Laajamies](#) | Washingtonin Isot Kirjaimet | [POISTA](#)
[Maalainen](#) | Jurvan Juntit | [POISTA](#)
[Maario](#) | Hampaattomat Kanadalaiset | [POISTA](#)
[Metsänen](#) | Stadin Ilmaveivit | [POISTA](#)
[MongoiIny](#) | Venäjän Mafia | [POISTA](#)
[Murmutti](#) | Team53 | [POISTA](#)
[nyrkkieki](#) | Laihian Alisuorittajat | [POISTA](#)
[Ole-Olof](#) | Norjan Kalastelijat | [POISTA](#)
[Oveski](#) | Washingtonin Isot Kirjaimet | [POISTA](#)
[Panda1](#) | Sneezing Pandas | [POISTA](#)
[Panda2](#) | Sneezing Pandas | [POISTA](#)



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Ylläpito | pelaajat

Luo tai muokkaa pelaajia. [Takaisin](#)

Nimi

Linkki

Joukkue

Pelipaikka

Pelinumero

Tallenna




UUTiset :: Tulokset :: Tilastot :: SÄÄNNÖT :: Tietoa :: Ylläpito

Ylläpito | joukkueet

Muokkaa joukkueiden tietoja tai [luo uusi joukkue](#).

[Takaisin](#)

[Berliinin Munkki](#) | [POISTA](#)
[Fortune Club](#) | [POISTA](#)
[Hampaattomat Kanadalaiset](#) | [POISTA](#)
[Hollywoodin Filmaajat](#) | [POISTA](#)
[HT Hokkarit](#) | [POISTA](#)
[Isän Maan Toivot](#) | [POISTA](#)
[Jurvan Juntit](#) | [POISTA](#)
[Laihian Alisuorittajat](#) | [POISTA](#)
[Norjan Kalastelijat](#) | [POISTA](#)
[Sneezing Pandas](#) | [POISTA](#)
[Stadin Ilmaveivit](#) | [POISTA](#)
[Team53](#) | [POISTA](#)
[Työttömät Hitsarit](#) | [POISTA](#)
[Venäjän Mafia](#) | [POISTA](#)
[Washingtonin Isot Kirjaimet](#) | [POISTA](#)



EHKL
ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTiset :: Tulokset :: Tilastot :: SÄÄNNÖT :: Tietoa :: Ylläpito

Ylläpito | joukkueet

Luo tai muokkaa joukkueita.[Takaisin](#)

Nimi

Lyhenne

Linkki

Peliasu - koti

Peliasu - vieras

Kuvaus

Tallenna




UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Ylläpito | uutiset

Muokkaa vanhoja uutisia tai [luo uusi uutinen](#).

[Takaisin](#)

- 24.10.2011 | [Runkosarjan päätös lähenee](#) | [POISTA](#)
- 11.9.2011 | [Pelikielto: yberpeelo](#) | [POISTA](#)
- 10.9.2011 | [Kausi 2011/1 käynnistynyt](#) | [POISTA](#)
- 5.9.2011 | [Joukkueiden kokoonpanot](#) | [POISTA](#)
- 4.9.2011 | [Tarkennus sääntöihin](#) | [POISTA](#)
- 1.9.2011 | [Kauden 2011/1 aikataulu](#) | [POISTA](#)
- 29.8.2011 | [Vastuuhenkilöiden nimitykset](#) | [POISTA](#)
- 28.8.2011 | [Karsintojen tulokset](#) | [POISTA](#)
- 23.8.2011 | [Liigan säännöistä](#) | [POISTA](#)
- 20.8.2011 | [Karsinnat](#) | [POISTA](#)
- 6.8.2011 | [Ilmoittautumiset](#) | [POISTA](#)
- 6.8.2011 | [EHKL perustettu](#) | [POISTA](#)



EHKL

ERINOMAINEN HIETALAN KIEKKO.TK-LIIGA

UUTISET :: TULOKSET :: TILASTOT :: SÄÄNNÖT :: TIETOA :: YLLÄPITO

Ylläpito | uutiset

Luo tai muokkaa uutisia.

[Takaisin](#)

Otsikko

Teksti

Tallenna